MICROHOBBY

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES SINCLAIR

ESPECIAL

Nº1 300 PTS

HOBBY PRESS, S.A.

HARDWARE

TODAS LAS IMPRESORAS SPECTRUM

COMO USAR ENSAMBLADORES



LAS MICROESTRELLAS MAS FAMOSAS DE LA PROGRAMACION

los Supercincuenta

Si vas a comprar un juego ¡CONSULTA ANTES ESTA GUIA!

Te ofrecemos dos lenguajes el "PASCAL" y el "C"

SUPERTRUCOS

Una relación de rutinas para que hagas superprogramas

Director Editorial
José I. Gómez-Centurión

Director de Números Especiales
Gabriel Nieto

Director de Microhobby
Domingo Gómez

Redactora Jefe
Africa Pérez Tolosa

Diseño
Rosa María Capitel, José Flores, Carlos Tejero

Redactor
Amalio Gómez

Colaboradores
Rafael Prades, J. M. Lazo, Heliodoro Martín, Alejandro Julvez, Marcos Ortiz, David Sapuerta

Fotografía
Javier Martínez, Carlos Candel

Portada
José María Ponce

Dibujos
A. Perera, F. L. Frontán
J. M. López Moreno, J. Igual, Enrique Almendros

Edita
HOBBY PRESS, S. A.

Presidente
María Andrino

Consejero Delegado
José I. Gómez-Centurión

Jefe de Publicidad
Marisa Esteban

Publicidad Barcelona
José Galán Cortés
Tels.: 303 10 22 - 313 71 76

Secretaría de Dirección
Marisa Cogorro

Suscripciones
M.ª Rosa González
M.ª del Mar Calzada

Redacción, Administración y Publicidad
La Granja, s/n
Polígono Industrial de Alcobendas
Telex: 49480 HOPR

Dto. Circulación
Carlos Peropadre

Distribución
Coedis, S. A. Valencia, 245
Barcelona

Imprime
ROTEDIC, S. A. Ctra. de Irun, km. 12,450 (MADRID)

Fotocomposición
Espacio y Punto, S. A.
Paseo de la Castellana, 268

Fotomecánica
GROF
Ezequiel Solana, 16

Depósito Legal
M-36.598-1984

Representante para Argentina, Chile, Uruguay y Paraguay, Cía. Americana de Ediciones, S.R.L. Sud América 1.532. Tel.: 21 24 64. 1209 BUENOS AIRES (Argentina).

Solicitado control OJD

MICROHOBBY ESPECIAL

ESPECIAL MICROHOBBY-AÑO I-N.º 1-NOVIEMBRE 1985

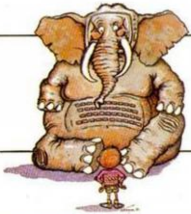
Esta es la primera vez que **MICROHOBBY** se hace especial *(y desde luego no será la última)* con la intención de ofrecer a sus lectores un número intemporal plagado de utilidades, que sea compendio y guía para todos sus seguidores, con actualidad y polémica, sin faltar, desde luego, el toque personal que nos caracteriza.

Esperamos haber conseguido este objetivo.

4 COLOQUIO — *El presente y el futuro de la microinformática visto por cinco especialistas en el tema.*



10 SUPERTRUCOS — *Una sección que recoge los mejores trucos publicados hasta el momento en MICROHOBBY.*



17 LOS SUPERCINCUENTA — *Balance de los cincuenta mejores programas que han caido en nuestras páginas desde el número 1.*

34 «COLEGA ELEFANTE» — *Un relato de Jesús Torbado.*

38 PROGRAMA — *El Cinquillo.*



42 GRAFICOS EN PASCAL

48 ESPECIAL IMPRESORAS — *Amplia relación de impresoras para Spectrum.*

64 LAS MICROESTRELLAS — *Todos los personajes más famosos del mundo del software.*

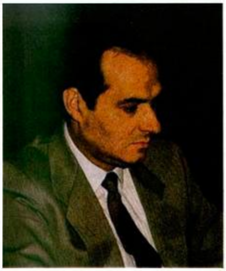


70 GENS-MONS — *Para ensamblar y desensamblar programas.*

76 EL «C» — *El otro lenguaje de programación.*

Presente y futuro

Muchos nos preguntamos por qué una evolución tan rápida y en tan breve espacio de tiempo ha tenido lugar en un país tan poco informatizado como ha sido el nuestro. Para hablar de este vertiginoso desarrollo y del futuro español en este terreno, hemos conseguido reunir a cinco cualificados representantes en el campo del ordenador personal: Ricardo García Gete, de Investrónica; Paco Pastor, de Erbe; Fernando Balairón, de ABC; José Villar, de Sinclair Store y un programador, Paco Suárez.



Ricaro García Gete



José Villar

De entre todas sus opiniones una ha sido realmente unánime, la sorpresa ante la rapidez de implantación del ordenador en España.

«Cuando empezamos —afirma García Gete— por el año 81, nadie podía esperar que de acuerdo al nivel tecnológico y al poder adquisitivo del país, se pudiera llegar al punto en que nos encontramos ahora. Pero fue un hecho. Hay países como Inglaterra que han quedado totalmente sorprendidos ante este desarrollo, no sólo de Sinclair sino de otras máquinas como Commodore o Amstrad.»

Se apunta como años álgidos en venta el 83 y 84, mientras que ya en 1985 se produce en España un gran bache, algunos piensan que por saturación, pero que se debió más bien a varias razones de tipo coyuntural que sujetaron el bolsillo del español. «Lo cierto —añade García Gete— es que estamos muy lejos de llegar a esta saturación si tenemos en cuenta las cifras, ya que hay un parque actual de medio millón de ordenadores».

Del Software y la Piratería...

A nivel de software el crecimiento no ha sido pequeño. «En este sentido —interviene ahora Paco Pastor— creo que Inglaterra se está dando cuenta de que aquí hay un mercado potencial importantísimo. Algo que antes era muy difícil de conseguir, una licencia de distribución y fabricación de software, ahora se obtiene racias a que ellos son los más interesados.»

Para Fernando Balairón la prudencia inglesa anterior no se debía a otra cosa que a la visión del poco volumen de ventas por la supremacía de la piratería en nuestro país. Lo que sí parece cierto es que no guarda ninguna relación con el consumo de ordenadores.

«Hay que tener en cuenta —afirma Paco Pastor— que en proporción al parque de Spectrum, las ventas de software son ridículas ya que hablar de quince mil unidades de programas no sería una locura si pensamos que hay medio millón de ordenadores. Sin embargo, esta cifra es impensable y lo es, fundamentalmente, por culpa de la piratería, tema que, por otro lado, se está clarificando bastante en los últimos días.»

Esas medidas «antipiratas» beneficiarían considerablemente el software, aumentando su calidad, como apuntaban Paco Portalo, y motivando una mayor proliferación de programadores, como insinuaba Paco Pastor. «Hay que tener en cuenta —afirma— que hoy por hoy no les es rentable programar, y no sólo por culpa de los piratas sino también por la gran cantidad de copias «por amistad» que salen de una cinta. Todos somos conscientes de que cada copia que vende un comercio genera de 5 a 8 copias más de amiguetes, por lo que a la postre, el parque estará cubierto pero sólo un 10% lo será por citas originales».

Un juguete llamado ordenador

Pero ¿qué ordenador prefiere el usuario español? ¿qué le lleva a decantarse por una u otra marca? Según Pepe Villar, en los orígenes de la comercialización el padre era el que se preguntaba cuál sería el más idóneo para su hijo. Hoy día es éste el que lo elige «y esta elección está a veces motivada por la posibilidad de acceder a un mayor número de copias de que ya dispone».

Por otro lado, el consumidor de hoy cuenta con un mayor asesoramiento y se diferencia en mucho de su muy cercano antecesor, «que compraba a tontas y a locas. Hoy el mercado está determinado por las marcas líderes, el Spectrum, que con la aparición del 128 se va a reforzar aún más; en segundo lugar está el Commodore y, como novedad, el Amstrad. La gente habla muy bien de él porque todavía no tiene un número suficiente en el mercado como para que empiecen a surgir las pegas. Todo esto hace que la gente hoy sea más selectiva».

> **“Los años álgidos de la venta fueron el 83 y el 84, pero ya en 1985 se produce un bache en España”**

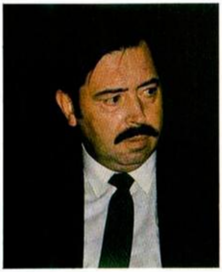
Una pregunta se hace indispensable a la hora de hablar del consumidor informático español ¿Con qué intención compra un ordenador, para jugar, para regalar o para otros usos?

«Yo —habla Pepe Villar— como distribuidor puedo afirmar que el negocio únicamente es rentable si se tiene en cuenta dos fechas claves: Reyes y vacaciones, ya que el 50% de la venta se realiza en estas dos épocas. Es justo decir que el ordenador sigue siendo un regalo propio de fin de curso y Navidad. Ahora bien, el que sea un regalo de estas fechas no quiere decir que haya que enmarcarlo en el tema juegos.»

de la informática



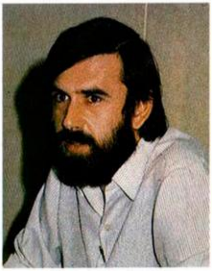
r



Fernando Balairón



Paco Pastor



Paco Portalo

Para Paco Pastor aquí, en este punto, es donde entra la psicología del niño. «El papá piensa que el ordenador es el tercer idioma, se lo compra al niño con esa ilusión y luego el hijo es el que decide y, por supuesto, prefiere jugar con él. Nosotros, como empresa de software nos hemos decantado por el juego porque alguna tímida incursión que hemos hecho en programas educativos no nos ha salido rentable. A pesar de ello, estas Navidades nos vamos a meter con él aunque nos da la sensación de que el niño nunca se va a comprar un programa educativo».

Si entramos de lleno en la psicología del niño, podremos comprobar que, por lógica, él nunca va a pedir a su padre un ordenador para jugar, se lo pedirá para aprender. Después... ya veremos...

«Hay que tener en cuenta —interviene Pepe Villar— que el juego es previo al aprendizaje porque después, si establecemos un baremo, de cada cien niños un 75% se queda en el juego y un 25% profundiza. De ese 25% que aprende programación, que se preocupa, a los dos o tres años ya demanda un producto de mayor capacidad. Puntualizando, creo que el ordenador doméstico ha servido de elemento de cultura que el padre ha comprado al hijo por que se lo ha pedido para jugar y sin embargo, el padre se lo ha comprado para aprender. A pesar de ello, hay que reconocer que el software educativo no se vende y, de alguna manera, esta corriente ha sido apoyada a través de todos los medios, en la importación, en la que el juego ha sido el más demandado e, incluso, las revistas especializadas hacen especial hincapié en el juego, lo que beneficia y daña al producto.»

La diversión como utilidad

Un factor que en un principio hemos pasado por alto, ha sido el porqué de este desarrollo vertiginoso del ordenador en España. Para García Gete esto tiene fácil explicación.

«Hay opiniones para todos los gustos,

> **“En proporción al parque de Spectrum, las ventas de software son ridículas”**

pero nosotros creemos que existe un punto muy importante: el padre español está acostumbrado a gastar dinero en la educación de su hijo intentando darle lo mejor y, por supuesto, una parte importante es la informática. Esto es lo que ha hecho subir el desarrollo del ordenador. Luego, hay que tratar de convencer al niño para que se inicie en la informática y la mejor manera de hacerlo es a través del juego», lo que por otro lado, podemos afirmar que se da en todo el mundo.

Un tercer punto a tener en cuenta se refiere al campo de las utilidades y en este sentido Fernando Balairón lo tiene muy claro. «La pequeña gestión, la utilidad, ha tenido en España mucha venta, si bien hay que decir que en este último año ha descendido considerablemente en favor del juego». Pero quizás a quien corresponde una opinión más objetiva como usuario y programador, sea a Paco Portalo. «Yo básicamente, estoy de acuerdo con lo que se ha dicho, aunque creo que es un poco falsa la idea. Aprender no puede aprender todo el mundo porque no todo el mundo tiene talento para ello. En cuanto al tema de la utilidad, veo que a un ordenador tipo Spectrum, las utilidades prácticas que uno puede sacar son muy pocas ya que no todo el mundo tiene necesidad de la pequeña gestión. Entonces tiene una utilidad básica, la de la diversión». Para Paco la supervivencia del micro pequeño pasa por buscar nuevas utilidades, muy relacionadas con las comunicaciones. En ese momento volverá a haber un boom del ordenador.

El ordenador ¿una panacea?

Una característica nacional, el querer comprar duros a peseta, y una promoción publicitaria muy determinada, han marcado en cierta medida el declive del microordenador. Y esto es fácil de entender. «Cuando una persona ha oído hablar de un ordenador —aclara Pepe Villar— su incultura o desconocimiento le ha llevado a creerse que aquello era la panacea y que lo hacía todo. Posteriormente, cuando ha comprobado que no era así, se ha decepcionado y se ha quedado en el juego. Efectivamente, como

AMPLIE SU SPECTRUM 48K



Nuevo periférico en
el Reino Unido.

¡Olvídate de la lentitud del cassette!

El **wafadrive** está especialmente diseñado para proporcionar máxima fiabilidad y alta velocidad de respuesta.
Y ademas...

¡GRATIS un «Spectral Writer» (Tratamiento de textos) y un cartucho virgen!
P.V.P.: 27.000 ptas.

Te regalamos antirrobo de vídeo marca Videolok valorado en 4.800 ptas.

WAFADRIVE

¡OFERTA SENSACIONAL!
WAFADRIVE +
R.A.T. + VIDEOLOK
¡POR SOLO!

28.500

R.A.T.

¡NO MAS CABLES: DOMINA TU SPECTRUM A DISTANCIA!
JOYSTICK DE CONTROL REMOTO PARA TU SPECTRUM 48 K o PLUS

- Compatible con todo el Software existente.
- Interface incorporado.
- P.V.P.: 5.900



Deseo recibir los periféricos señalados con una X.

WAFADRIVE ☐
R.A.T. ☐
WAFADRIVE + R.A.T. + VIDEOLOK ☐

NOMBRE
DIRECCION
POBLACION D. P.

PAGO TALON NOMINATIVO ☐
GIRO POSTAL ☐

JOBISA: C/ VERGEL, 8. DENIA (ALICANTE)
TELF.: (965) 78 51 11 - 78 50 69

Firma

decía Fernando, ha habido programas en el año 1984 que salieron para gestión y hay que preguntarse qué gestión se lleva en un hogar medio español. Yo creo que ninguna. Son una élite muy minoritaria que sí encuentra en el Spectrum un buen instrumento; pero por desgracia son muchos los que piensan que con este ordenador y un buen programa de gestión pueden hacer maravillas y se olvidan que el programa no lo hace todo y que es imprescindible saber contabilidad. Ahí viene la frustración».

Un futuro prometedor

Basándonos en la experiencia acumulada durante estos años de expansión informática, podemos decir que el futuro del ordenador está en una mayor «practicidad» a nivel de aplicación que es, en definitiva, lo que busca el usuario para encontrar una total utilidad al producto.

Pepe Villar puntualiza sobre el tema. «¿Qué cuál es el futuro? Pues bien, se ha producido una gran inquietud en este país por todo lo relacionado con la informática, entonces, afortunadamente, hay un colectivo muy grande que todavía no tiene ordenador y ese colectivo, basado en esa corriente experimentada a lo largo de tres años, cuando se lo compre se irá a los ordenadores domésticos. Luego, las personas que ya están bautizadas a lo largo de tres años, que ya han tenido ordenador y han recibido el efecto frustración de los duros a pesetas, esta gente demanda un ordenador superior que si pudiera gastarse un millón de pesetas, les llevará a un IBM PC de entrada porque sabe que lo va a tener todo; pero si se queda en las cien mil, encuentra el sucedáneo en donde entra el ordenador con unidad de disco incorporada. Por último, están los profesionales que van a ir a por el ordenador de ciento cincuenta mil pesetas a quinientas mil. El futuro va a ir por ahí, pero forzosamente pasa porque la corriente que se ha despertado a lo largo de estos últimos años va a dar una cantidad de venta brutal en el primer ordenador.»

Para García Gete el mercado no está ni mucho menos saturado y por tanto el futuro del ordenador y de la informática a nivel ventas estará, por una parte, en los usuarios que sigan comprando el home computer según su poder adquisitivo y, por otra, los que quieran ir a otra cosa mayor, aquellos interesados en la informática, y ahí es donde tiene que haber un ordenador que efectivamente tenga validez y múltiples aplicaciones a nivel profesional o de comunicaciones.»

La controversia de los «Ks»

Otro punto de influencia a la hora de adquirir un microordenador es su capacidad de memoria, algo demasiado valorado a nivel de usuario. 16, 48, 64 y ahora la afluencia masiva de los 128 están saturando el mercado pero, ¿es realmente necesaria? ¿Para qué quiere el usuario más memoria? «A nivel hogar —afirma García Gete— con los 48 o 64 K es más que suficiente, lo que ocurre es que el mercado ha ido pidiendo más y más y, sobre todo, los fabricantes han ido ofreciendo más (quizás porque dar hoy 128 K cuesta lo mismo que hace algún tiempo ofrecer el 16 K, por ejemplo) sobre todo por razones tecnológicas; pero desde luego no es necesario».

Si bien esto es cierto, no lo es menos el hecho de la existencia por parte del usuario de una mentalidad muy definida: caballo grande ande o no ande, en este caso, mayor número de Ks sean necesarios o no, mentalidad que deben en gran medida a la falta de una información correcta al respecto por parte de distribuidores y fabricantes.

«Bueno, en este sentido —afirma Paco Portalo— creo que cuanto mayor memoria mayor será su utilidad, según los casos. Pensando en el juego, por ejemplo, cuanto mayor sea la memoria mejores gráficos se conseguirán mejorando considerablemente el programa.»

«En este sentido —interrumpe García Gete— hay que tener en cuenta la evolución tecnológica. Hace tres o cuatro años no había pastillas de 64 K y hoy día vale lo mismo que antes las de 8. Entonces al fabricante le da lo mismo meterla de 16 o 64 y, naturalmente, aboga por estas últimas. Sobre todo teniendo en cuenta que son más requeridas por el usuario, con un mínimo aumento de costas.»

En este punto cabe preguntarse si realmente a mayor número de Ks corresponde una mayor calidad de programa, lo que no siempre es cierto, pero sí es válido ante la falta de información e incultura informática que caracteriza al español medio.

En este sentido —afirma Paco Pastor— creo que tenemos un desconocimiento total, y llegamos a pensar que cuanto más memoria tenga nuestro ordenador mayores cosas podremos hacer con él. Quizás en un futuro no muy lejano el conocimiento informático nos lleve a discernir con mayor objetividad».

«Eso está claro, desde luego —añade Pepe Villar— porque yo desde mi punto de vista puedo asegurar que son muy pocos los que han oído hablar del microprocesador, por ejemplo. En este sentido, uno de los ordenadores que mejor está, en cuanto a la relación calidad precio, es el QL y si la gente tuviera el mínimo conocimiento de su microprocesador, se vendería mucho más. Hay que reconocer que el impacto en la venta está en el número de Ks.»



“El 50% de las ventas se realizan en vacaciones y Reyes”

«Es muy difícil establecer comparaciones entre un ordenador y otro —interrumpe Paco Portallo— ya que hay muchos factores que intervienen. Personalmente, como técnico, puedo decir, refiriéndonos nuevamente al QL, que no le han sacado todas las prestaciones o capacidades que el micro podría tener...»

La alternativa del ocio

En la andadura del coloquio nos transportamos a la tercera edad, a la existencia del ocio y en este terreno, tiene mucho que decir y hacer el microprocesador.

«En este campo —afirma Pepe Villar— es cada vez más grande la demanda en profesiones liberales como por ejemplo, médicos, notarios, ya jubilados que profundizan con el Spectrum hasta límites insospechados. La gente mayor encuentra en el ordenador un instrumento muy útil para llenar su tiempo de ocio...»

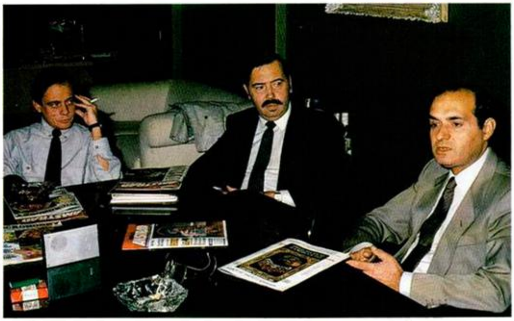
Extrapolando situaciones, otra cultura del ocio, la de la gente joven, tiene mucho que decir en esto y en torno a ellos se desarrolla el ordenador como juguete que va a desbancar a cualquier otro. En este sentido son muchos los muchachos que quieren llegar a más, aprender a programar, pero ¿cómo pueden llegar a hacerlo?...

La opinión casi unánime de los reunidos apunta al gran reto del chico por investigar, destripar y conocer lo que tiene entre sus manos, sin descartar la ilusión de poder llegar a ser como su ídolo, el autor de tal o cual programa. De entre todos ellos, algunos pocos lo conseguirán aunque eso les lleve a aprender cualquier idioma informático.

«De todo esto surge la picaresca en el muchacho que deja de ser sana cuando se da cuenta que puede sacar dinero de aquello que ha hecho —afirma Pepe Villar—. Es decir, cuando descubren que lo que les ha servido de aprendizaje les sirve para ganar dinero. El siguiente paso es ir a venderlo al Rastro, lo que empieza ya a ser preocupante vendiendo los mejores temas a trescientas pesetas y encima desprotegidos...»

¿Coleccionistas de software?

En España es un hecho la gran rapidez con que vienen y se van los grandes éxitos ingleses o americanos, su poca permanencia en «cartelera» cuando en su país de origen perduran meses y meses. ¿Cabría pensar que existe un cierto afán de coleccionismo en torno al software?



“La vida de un programa en España es bastante más larga que en Inglaterra”

Para Paco Pastor ocurre todo lo contrario. «Creo que la vida de un programa en España es bastante más larga que en Inglaterra y hay títulos que se están vendiendo durante un año... Lo que sí sorprende un poco es la avidez del mercado, que está creciendo en una proporción muchísimo más rápida que en otros países, aunque hemos llegado tarde y nos hemos perdido una serie de pasos que quizás es lo que haya hecho que la gente en otros países esté un poco más educada. Ahora, también eso es un poco la psicología del español, puede que llegue tarde pero cuando llega lo hace con más ansias que ninguno».

«Lo que sí hay —afirma Fernando Balairón— es un afán de coleccionismo por parte de todos los chavales y se llegan a encontrar hasta con sesenta cintas que, por supuesto, no han comprado y que no les ha dado tiempo a ver, se las pasan y las copian.»

«A parte de esto —interviene Pepe Villar— habría que ver en qué tiempo se quemaba un programa en el año 1984 y en qué tiempo se quema ahora. Creo que tiene la culpa la sociedad de consumo porque si antes siempre el niño ha tenido el afán de ser el único en tener el *primer juego que acaba de aparecer en* Londres, ahora está aquí todo y todo lo que aquí se publica inmediatamente va a por ello y se lo consiguen aunque no hayan visto el anterior que tienen. Se les está despertando un hambre increíble...»

En toda esta cuestión merece tenerse en cuenta el precio del software. ¿No es demasiado caro para el ritmo de consumo existente?

«No, ni mucho menos —apunta Paco Pastor— pienso que la relación entretenimiento software es lo más barato que existe en el mercado. Lo cierto es que ante la abalancha existente el chico tiene que hacerse selectivo. Pero volviendo a su bajo precio puedo decir, por ejemplo, que sale más barato que ir al cine o alquiler una película que consumes en dos horas. El software te ofrece la posibilidad de divertirte con un programa incluso meses hasta que consigues completarlo. Luego, es rentable.»

La opinión general no parece ser la misma y para Pepe Villar, los muchachos consiguen como sea el dinero para comprar la última cinta en el mercado «y como auténticos zombis, que lamentablemente son, se colocan ante el ordenador todo el día hasta destriparla. Al día siguiente ya es viejo el programa».

Muchas opiniones más: que es culpa de la sociedad de consumo en que vivimos, de los mismos programadores que no cesan de producir, hasta afirmaciones como que podría ser infinitamente más barato si se acabara con el tema de las copias, pasando por las ideas de considerar al ordenador un enemigo en potencia por estar convirtiéndose en un juguete indispensable para el muchacho, estas y otras más, fueron vertidas a lo largo de este largo coloquio que, creemos, ha dejado claro el punto de vista del pensamiento especializado. Eso esperamos.

¡NUEVO!

SIEMPRE LOS PRIMEROS EN TENER LO ULTIMO

circulo de soft
MICROAMIGO S.A.

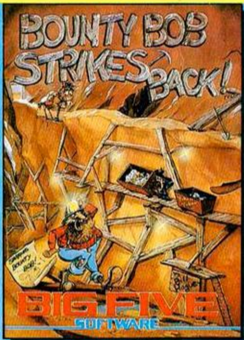
P.° de la Castellana, 268, 3.° C. 28046-MADRID.
Tel.: (91) 733 25 00



FIGHTING WARRIOR

Para salvar a la princesa encerrada en la Gran Pirámide, tendrás que enfrentarte con tu espada a criaturas infernales que tratarán de impedírtelo por todos los medios. Este juego viene avalado por la firma de los creadores de Exploding Fist.

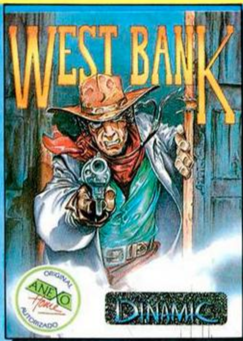
P.V.P.: 2.100 ptas.
Precio Socios C. de Soft: 1.890 ptas.



BOUNTY BOB

Junto con el Manic Miner, éste es el mejor juego de «plataforma» aparecido para ordenador y cuyo éxito en Commodore va a repetirse ahora en su versión para Spectrum.

P.V.P.: 2.100 ptas.
Precios Socios C. de Soft: 1.890 ptas.



WEST BANK

Defiende el banco de Soft City del ataque de los forajidos y consigue escribir tu nombre en la leyenda del «FAR WEST».

P.V.P.: 2.100 ptas.
Precio Socios C. de Soft: 1.890 ptas.

¡¡¡...Y LOS TRES PROGRAMAS POR SOLO 4.990 PTAS!!!

¡HAZTE HOY MISMO SOCIO DEL CIRCULO DE SOFT! Además de poder adquirir tus programas al mejor precio, recibirás información de forma periódica y gratuita, del mejor software que aparezca en el mercado.

¿QUE HAY QUE HACER PARA SER SOCIO DEL CIRCULO DE SOFT? Así de fácil: envíanos por correo tu nombre, dirección y modelo de ordenador, o bien, pide por teléfono o por correo tu primer programa. ¡Y entrarás a formar parte del CIRCULO DE SOFT de forma inmediata!

☐ SÍ, quiero ser SOCIO desde hoy mismo del CIRCULO DE SOFT y recibir periódicamente información de novedades de software, así como beneficiarme desde hoy mismo de los precios reducidos reservados a los SOCIOS y de sus Ofertas Especiales. El ser SOCIO no me obliga a compra alguna.

Si prefieres formalizar tu compra por teléfono puedes hacerlo llamando al (91) 733 25 00. ¡¡NO SE COBRAN LOS GASTOS DE ENVIO POR CORREO!!

TITULO	P.V.P.	ORDENADOR

☐ Contrarreembolso ☐ Giro Postal ☐ Talón adjunto a Microamigo, S.A. ☐ Tarjeta VISA n.° ____ Fecha caducidad ____

Nombre ____ Apellidos ____ Edad ____

Domicilio ____ Teléfono ____

Localidad ____ C.P. ____ Provincia ____

SUPERTRUCOS

SEGURIDAD EN LAS COPIAS

Para muchos usuarios del microdrive y la interface 1, con frecuencia es un problema tratar de obtener copias de seguridad de sus programas favoritos en cinta de cassette, bien sean de juegos o de utilidad, debido a que estos, normalmente escritos en lenguaje máquina, «pisan» la zona que el sistema operativo del Spectrum reserva para las operaciones que debe realizar con el microdrive.

En efecto, la ROM «fantasma» de la interface 1, cuando entra en servicio reserva un buffer de regular tamaño para acomodar sus propias variables de sistema, y una memoria intermedia pare realizar las operaciones de entrada/salida; cualquier comando de los tipos OPEN, MOVE, VERIFY, LOAD y, para el caso que nos ocupa, SAVE que afecte al microdrive, necesita como mínimo 595 bytes libre a partir de una zona determinada de la memoria.

Si el programa que queremos pasar a microdrive ocupa esta zona de memoria, para solucionarlo debemos conocer tres cosas:

A) Dirección en la que carga el lenguaje máquina del programa.

B) Longitud de dicho código máquina.

C) Dirección a reubicar, la cual elegiremos nosotros.

De cada uno de estos datos deberemos obtener un byte alto y un byte bajo, con el fin de dárselos como parámetros al programita en lenguaje máquina que vamos a indicar ahora, debido a J. Antonio García Boal, y que resolverá el problema. Se obtienen de la manera siguiente:

byte alto = INT (dato/256) (FORMULA 1)

byte bajo = ((dato/256) – byte alto) • 256 (FORMULA 2)

Una vez apuntados los tres datos obtenidos de esta forma, teclearemos el siguiente programa, que no es más que un cargador Basic, como siempre:

```
10 FOR N=60000 TO
   60011: READ A: PO-
   KE N,A:NEXT N
20 DATA 33, BYTE BAJO
   DE A), BYTE ALTO
   DE A)
30 DATA 17, BYTE BAJO
   DE C), BYTE ALTO
   DE C)
40 DATA 1, BYTE BAJO
   DE B), BYTE ALTO DE
   B), 237, 176, 201
50 LET V = USR 60000
```

en donde las palabras byte alto y byte bajo deben ser sustituidas por los valores correspondientes obtenidos del empleo de las fórmulas 1 y 2.

Vamos a ver todo esto con un ejemplo: supongamos que tenemos un bloque de código máquina cuya dirección original de carga es la 23600 y ocupa 10.000 bytes.

Elegimos, por ejemplo, la dirección 30.000 como dirección de carga de momento y tecleamos LOAD " " CODE 30.000. A continuación, lo salvamos en cartucho con la orden.

SAVE • "M"; 1; "NOMBRE" CODE 30000,10000

Usando las fórmulas 1 y 2 calcularíamos los valores de los bytes alto y bajo, sustituyéndolos en el programa anterior en las líneas 20-40, las cuales quedarían así:

```
20 DATA 33, 48, 117
30 DATA 17, 48, 92
40 DATA 1, 16, 39, 237,
   176, 201
```

y salvamos el programa Basic en cartucho. Lo ejecutamos, y si nuestro programa en máquina, que todavía permanece en la memoria, lo requiere, hacemos RANDOMIZE, USR, DIRECCION DE ARRANQUE.

CARGADOR HEXADECIMAL EN CODIGO MAQUINA

Una vez más el lenguaje máquina viene en nuestro auxilio para implementar una pequeña rutina de gran utilidad para los programadores, y que puede emplearse desde Basic sin ningún problema.

Se trata de representar en hexadecimal cualquier número decimal, de 0 a 65535.

Para ello, como puede observarse en la línea número 20 del programa cargador Basic, introducimos nuestro número decimal en la variable SEED, mediante la instrucción RANDOMIZE.

Luego, basta llamar a la rutina en máquina que hace el trabajo duro e instantáneamente aparecerá el número en la pantalla.

Como casi siempre, proponemos como dirección de ensamblado el buffer de impresora, utilizado en el programa cargador.

Para los que posean un ensamblador o estén interesados en averiguar cómo funciona la rutina, proporcionamos también la rutina escrita en lenguaje ensamblador.

```
 10            ORG   23296
 20  ;
 30            RES   0,(IY+2)
 40            LD    DE,(23670)
 50            LD    C,D
 60            XOR   A
 70            CP    D
 80            JR    Z,BAJO
 90            CALL  HEXA
100  BAJO      LD    C,E
110            CALL  HEXA
120            LD    A,13
130            RST   16
140            RET
150  HEXA      LD    A,C
160            AND   #F0
170            SRL   A
180            SRL   A
190            SRL   A
200            SRL   A
210            CALL  LOW
220            LD    A,C
230            AND   #F
240  LOW       ADD   A,"0"
250            CP    58
260            JR    C,PR
270            ADD   A,7
280  PR        RST   16
290            RET
```

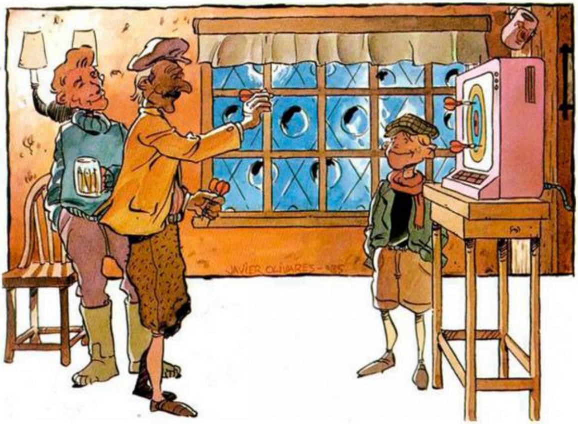
```
  10 INPUT "NUMERO ";CIF
  20 RANDOMIZE CIF: RANDOMIZE US
R 23296
  30 GO TO 10
 100 FOR N=23296 TO 23346: READ
A: POKE N,A: NEXT N
  110 DATA 253,203,2,134,237,91,
118,92,74,175,186,40,3,205,24,91
,75,205,24,91,62,13,215,201,121,
230,240,203,63,203,63,203,63,203
,63,205,41,91,121,230,15,198,48,
254,58,56,2,198,7,215,201
```

PARA DIBUJAR COMO QUIERAS

Cambiando los números del FOR, o haciendo operaciones en las coordenadas de los PLOT/DRAW, o bien introduciendo nuevos bucles, conseguiremos infinidad de dibujos, según nos indica E. Sánchez García.

```
 5 FOR a=0 TO 100
10 PLOT 90,a: DRAW 90,a: NEXT
```

INPUT EN CUALQUIER LUGAR DE LA PANTALLA



En algún programa que diseñemos, nos puede interesar, aunque sea por razones puramente estéticas, realizar un INPUT en cualquier parte de la pantalla.

Para poder llevar esto a cabo, nos vemos obligados a volver sobre una variable del sistema que ha aparecido a menudo en esta sección de trucos.

Nos referimos a DEF_SZ, localizada en la posición de memoria 23659.

Como recordaréis, el valor almacenado en esta posición le «dice» a la ROM del Spectrum el número de líneas de la pantalla, comenzando por la parte inferior de la misma, que debe dedicar a la ventana de mensajes y comandos.

En efecto, cuando hacemos un INPUT en medio de un programa, el cursor aparece en las dos líneas inferiores de la pantalla.

Para solucionarlo, en principio la respuesta parece clara: puesto que DEF_SZ almacena el número de líneas de la ventana inferior, pokeemos allí y démosle el valor 24, es decir, toda la pantalla.

Por desgracia, el asunto no resulta tan sencillo. Cuando se ejecuta un comando INPUT, lo primero que hace la rutina de la ROM es volver a asignar a DEF_SZ el valor 2, con lo que nuestro POKE se esfuma de dicha variable y el INPUT se muestra donde siempre.

La solución definitiva está, una vez más, en el lenguaje máquina.

Cuando el ordenador se conecta, el programa de inicialización carga el par de registros IY con la dirección de comienzo del área de variables del sistema. Esto permite al programador acceder a cualquiera de ellas mediante lo que se conoce como «direccionamiento indexado». DEF_SZ se encuentra en IY+49.

Sólo necesitamos dos minúsculas rutinas en ensamblador, una para asignar a DEF_SZ el valor 24 (toda la pantalla, rutina número 1) y la otra para restituir el valor original, 2 (rutina número 2).

Estas dos rutinas son reubicables, esto es, pueden colocarse en cualquier posición de la memoria. Se sugiere el buffer de impresora, pero, por si esa dirección no interesa, aquí tenéis un pequeño cuadro con el código de operación (los números que hay que pokear en la memoria) y los mnemónicos:

RUTINA NUMERO 1

Código de operación

253 54 49 24
201

Mnemónico

LD (IY+49),24
RET

RUTINA NUMERO 2

Código de operación

253 54 49 2
201

Mnemónico

LD (IY+49),2
RET

Supongamos que las rutinas 1 y 2 han sido ubicadas en las direcciones «DIR_RUT_2», respectivamente.

Necesitamos ahora un programa Basic que complete a estas dos rutinas, y que las llame en el momento adecuado. Sería el siguiente:

```
10 LET ABRE=DIR_RUT_1: LET CIERRA=DIR_RUT_2
20 FOR I=ABRE TO ABRE + 4:READ X:POKE I,X: NEXT I
30 FOR I=CIERRA TO CIERRA + 4:READ X:POKE I,X:NEXT I
40 INPUT "" AND USR ABRE; AT X,Y; N$; "" AND USR CIERRA
50 DATA 253,54,49,24,201
60 DATA 253,54,49,2,201
```

en donde:

la línea 10 inicializa las variables ABRE y CIERRA a las direcciones donde se supone que las rutinas 1 y 2 han sido colocadas. Estas direcciones tenéis que decidirlas vosotros y sustituir los números que hayáis escogido por «DIR_RUT_1» y «DIR_RUT_2». Insistiendo en que estos valores pueden ser los que queráis, os sugerimos 23296 para la rutina 1 y 23231 para la rutina 2, con lo que ABRE vale 23296 y CIERRA 23231.

La línea número 40 involucra varios trucos:

1. INPUT "" borra las dos líneas inferiores de la pantalla.

2. INPUT "" AND USR ABRE además, llama a la rutina número 1, la cual asigna a DEF_SZ toda la pantalla.

3. X,Y son las coordenadas de la pantalla donde aparecerá el cursor del INPUT, y que tenéis que sustituir por los valores que elijáis.

4. "" AND USR CIERRA llama a la rutina número 2 y devuelve a DEF_SZ su valor original.

Las líneas 20 y 30 cargan en memoria los bytes de ambas rutinas.

LAS SIETE LLAVES

Uno de nuestros lectores de Barcelona, Carlos González, nos manda un auténtico cóctel de trucos para la protección de programas, consiguiendo alejar nuestros listados de miradas curiosas.

El método es el siguiente:

1. encabezar el programa con una línea que diga POKE 23613,Ø

2. continuar con la segunda línea diciendo SAVE «nombre» CODE 23552,long en donde «long» es la longitud de nuestro programa

3. la tercera línea será INPUT «introduce clave»; LINE a$: IF a$ < > «clave» THEN RANDOMIZE USR Ø

Vamos a tratar de explicar lo que sucede en el ordenador cuando estas tres sentencias se ejecutan.

1. La posición de memoria 23613 corresponde a la variable del sistema ERR SP y, en ella, se almacena una dirección a cuyo contenido salta el microprocesador cuando se detecta un error Basic; para verlo con un ejemplo, supongamos que ERR SP contiene el número 31996, el cual hemos averiguado mediante la sentencia PRINT PEEK 23613 + 256*PEEK 23614. Supongamos también que repetimos la misma operación con este número, es decir, PRINT PEEK 31996 + 256*PEEK 3197 y obtenemos 4867. Esta es la dirección a la que saltará el ordenador cuando un error sea detectado. Por tanto, si colocamos en 23613 un cero, estaremos alterando la dirección de salto y, vaya usted a saber dónde irá la CPU cuando la condición de error sea detectada; se producirá el temido «system crash».

2. Lo que efectuamos aquí es grabar en cinta el programa Basic, como si se tratara de Bytes, con la salvedad de que también grabamos las variables del sistema íntegras. Esto implica que, al volver a cargar nuestro programa como LOAD "" CODE, éste se autoejecutará, ya que el valor de algunas de estas variables le impelen a ello.

3. Esta última, requiere menos explicación, ya que el INPUT LINE es conocido por todos nuestros lectores; algunos de ellos nos diraán que de un INPUT LINE se puede salir pulsando CAPS SHIFT + 6 y detener el programa; bien, es cierto a medias, ya que esto provocaría un salto a la famosa ERR SP que hemos alterado previamente; es decir, bloqueo de la máquina garantizado.



CAMBIAR LOS ATRIBUTOS DE LA PANTALLA

Proponemos una corta rutina en código máquina que nos permite cambiar instantáneamente el color del borde, papel y tinta a los valores que elijamos; la rutina se presenta en forma «artesanal», es decir, hay que construir el valor del byte de color y luego introducirlo mediante POKE o bien cambiar el valor en las DATAS.

Si se observa el listado del progama se verá que el número 41 se repite 2 veces; éste es el byte de color que indica papel 5 (cyan) y tinta 1 (blue).

El byte se construye multiplicando el valor del papel por 8 y sumándole la tinta [41=(5•8)+1].

Por ejemplo, para poner papel negro y tinta amarilla, cambiaríamos en las DATAS el valor de 41 por 6[6=(Ø•8)+6].

De paso, hemos incluido unos pocos bytes más que colocan el borde del mismo color del papel, empleando la instrucción RRCA con objeto de mover el número que representa el papel a los bits Ø, 1 y 2 del acumulador. Acto seguido, mediante la instrucción OUT (C), A cuyo equivalente ya vimos en un truco BASIC, conseguimos el efecto deseado.

```
  10 FOR N=40960 TO 40960+22: RE
AD X: POKE N,X: NEXT N
  15 LIST
  20 RANDOMIZE USR 40960
 100 DATA 33,0,88,54,41,17,1,88,
1,255,2,237,176,14,254,62,41,15,
15,15,237,121,201
```

Los colores no quedan fijados de forma permanente, así que si imprimimos algunos caracteres después, sin indicación esplícita de papel y tinta; aparecerán con los atributos que el ordenador conserve por defecto.

SIMULACION DE LA INSTRUCCION POP

Una de las utilidades del lenguaja Basic que el Spectrum no posee, es la sentencia POP. Este comando tiene la función de impedir que determinada subrutina, que llamemos mediante la sentencia gosub, retorne al programa principal; la utilidad

```
  10 REM ****************************
  20 REM      Esta rutina debe
  30 REM    colocarse fuera del
  40 REM       sistema Basic
  50 REM ****************************
  60 INPUT "Direccion de ensambl
ado? ";dir
  70 FOR n=dir TO dir+23: READ a
: POKE n,a: NEXT n
  80 DATA 237,123,61,92,59,59,19
3,225,209,122,254,62,202,54,31,5
9,227,235,237,115,61,92,197,201
```

de esta función resulta un tanto oscura y, como siempre, se comprenderá mejor practicándola. Observe que nos permite saltar desde una subrutina a cualquier parte del programa principal.

La rutina se ha escrito en lenguaje máquina y es reubicable, es decir, puede ejecutarse en cualquier parte de la memoria. En principio, se sugiere emplear el buffer de impresora para colocarlo, aunque debe tener en cuenta que si emplea comandos relacionados con la impresora, la rutina se corromperá.

Hemos incluido, someramente comentado, el listado en ensamblador y, para aquellos que prefieran utilizarlo directamente, un cargador Basic que solucionará el programa.



A. PERERA

```
 ; SUBRUTINA RELOCALIZABLE
 ;
         LD    SP,(23613)
         DEC   SP
         DEC   SP
 ;
 ; PUNTERO DEL STACK SE LE
 ; ASIGNA LA DIRECCION DE
 ; RETORNO DE ERROR -2
 ;
         POP   BC
 ;
 ; TOMA LA DIRECCION DE
 ; RETORNO DE LA ROM QUE
 ; TRATA LA SIGUIENTE
 ; SENTENCIA BASIC
 ;
         POP   HL
 ;
 ; TOMA DORECCION RETORNO
 ; DE ERROR
 ;
         POP   DE
 ;
 ; TOMA EL ULTIMO DATO DE
 ; LA PILA DE GOSUB
 ;
         LD    A,D
         CP    62
         JP    Z,#1F36
 ;
 ; MENSAJE DE "RETURN
 ; without GOSUB" SI ES
 ; FINAL DE PILA
 ;
         DEC   SP
         EX    (SP),HL
         EX    DE,HL
         LD    (23613),SP
 ;
 ; RESTAURA EL PUNTERO DE
 ; ERROR
 ;
         PUSH BC
 ;
 ; REPONE EN EL STACK LA
 ; DIRECCION DE RETORNO DE
 ; LA ROM
 ;
         RET
```

SUPERTRUCOS



SIMULACION DE LA SENTENCIA PRINT USING

Hemos recibido consultas de algunos lectores sobre cómo podrían formatear la salida impresa en la pantalla para conseguir, por ejemplo, una columna alineada de números para programas de aplicación técnica o utilidad.

José María Martínez Arbex nos ha resuelto el problema al enviarnos una pequeña rutina Basic que imita, hasta cierto punto, a la famosa y potente sentencia PRINT USING de otros dialectos de Basic.

El trabajo lo realiza la función definida en la línea 100; necesita dos datos: el núnero a representar y la longitud del campo donde va a ser representado.

Para flexibilizarla al máximo, hemos definido una variable, LONGCAMPO, inicializada a 15, que nos permite elegir la longitud máxima del campo de representación que queremos.

También está incluida una subrutina para atrapar errores (que el campo sea cero o que su longitud sea menor que la del número a pintar).

```
  10 REM *** SIMULACION DE ***
  20 REM ***   PRINT USING  ***
  30 :
  40 REM *** INICIALIZACION ***
  45 :
  50  LET SPACE=32: LET E$=""
  60  LET LONGCAMPO=15
  62  LET ERROR=0
  63  LET COMPROBAR=1000
  65 :
  70   FOR I=1 TO LONGCAMPO
  80     LET E$=E$+CHR$ (SPACE)
  90   NEXT I
  95 :
 100  DEF FN U$(N,L)=
       E$( TO L-LEN STR$ N)+
       STR$ N
 110 :
 120 REM *** ENTRADA DATOS ***
 125 :
 130   INPUT "NUMERO ";NUM
 140   INPUT "CAMPO ";LCAMPO
 145   GO SUB COMPROBAR
 147   IF ERROR THEN LET ERROR=
       NOT ERROR: GO TO 130
 148 :
 150   PRINT INVERSE 1;
        FN U$(NUM,LCAMPO)
 155 :
 160 GO TO (130 AND NUM)+
      (2000 AND NOT NUM)
 170 :
1000 REM *** COMPROBAR  ***
1010 REM ***  ERRORES ***
1020  LET ERROR=NOT LCAMPO OR
           (LEN STR$ NUM>LCAMPO)
1030 RETURN
```

COMO UN PIANO

Con este truco que nos ha mandado José Ignacio Rodríguez Valladolid, podrás convertir tu Spectrum en un elemental piano, donde «Q» será igual a «DO», «2» a «Do» #, «W» a «RE», «3» a «RE» #, «E» a «MI», «R» a «FA», «5» a «FA» #, «T» a «SOL», «6» a «SOL» #, «Y» a «LA», «7» a «LA» #, y «U» a «SI». Se basa en las funciones INKEY$ y BEEP. Para finalizar, apretar el número 1.

```
  10 LET duracion=1: LET tono=0
  20 PAUSE 0
  30 LET tono=(11 AND INKEY$="u"
)+(10 AND INKEY$="7")+(9 AND INK
EY$="y")+(8 AND INKEY$="6")+(7 A
ND INKEY$="t")+(6 AND INKEY$="5"
)+(5 AND INKEY$="r")+(4 AND INKE
Y$="e")+(3 AND INKEY$="3")+(2 AN
D INKEY$="w")+(1 AND INKEY$="2")
+(0 AND INKEY$="q")
  40 IF NOT tono AND INKEY$<>"q"
 THEN STOP
  50 BEEP duracion,tono
  60 GO TO 20
 RETURN L
```

PARA ACENTUAR Y OBTENER LA Ñ

Como habrás «sufrido», por experiencia, la falta de la «Ñ» y de la acentuación en los textos de los programas es un hecho. Por este motivo y para conseguir un perfecto castellano en ellos, José Luis González Sendra nos ha enviado un truco que consiste, precisamente, en esto: añadir los acentos y la ñ, mediante este pequeño listado. Como ha utilizado los GDU, nos manda también sus equivalentes.

```
   1 FOR n=0 TO 47: READ a: POKE
 USR "a"+n,a: NEXT n: DATA 4,8,5
6,4,60,68,60,0,4,8,56,68,120,64,
60,0,8,16,0,48,16,16,56,0,4,8,56
,68,68,68,56,0,8,16,68,68,68,68,
56,0,56,0,120,68,68,68,68,0
```

A	B	C	D	E	F
á	é	í	ó	ú	ñ

64 COLORES

A pesar de que el Spectrum cuenta con sólo 8 colores disponibles, nada nos impide combinarlos entre sí manera que a simple vista dé la sensación de tratarse de un nuevo color. Estos nuevos colores pueden usarse como gráficos, fondos, etc. y para su elaboración se precisan varias fases. Vayamos por partes. En primer lugar hemos de definir una rejilla a la que asignaremos distintos colores de tinta y papel. De este forma la mezcla de colores es más perfecta.

Introduzca las siguientes instrucciones:

Una vez ejecutado el programa, usted probablemente se sorprenderá al ver que todo se borra. Da la impresión de que algo ha fallado. Sin embargo, si el programa estaba correctamente introducido, todo ha ido como se esperaba. Para cerciorarse escriba: PRINT CRH$ 144 seguido de ENTER y verá aparecer en la esquina superior izquierda un pequeño recuadro parecido a un tablero de ajedrez en miniatura.

A continuación, teclee el segundo programa y verá, al hacerlo funcionar, cómo en pantalla le son mostradas las 64 posibles combinaciones de colores, debajo de cada una de las cuales aparecen dos números. El primero se refiere al color del papel, mientras que el segundo es el color de la tinta. Así, si usted desea utilizar el color 75, sólo tiene que escribir PRINT PAPER 7; INK 5; CRH$ 144 e inmediatamente aparecerá un cuadradito del color seleccionado.

Si quiere incorporar este trucod a sus programas, utilizará únicamente el primero, el segundo es sólo una demostración. ¡Ah! y no olvide quitar el NEW de la instrucción 3O, pues de lo contrario los efectos serían catastróficos para su recién estrenado programa.



PROGRAMA 1

```
  10 FOR N=0 TO 7: READ X
  20 POKE USR "A"+N,X: NEXT N
  30 NEW
  40 DATA 85,170,85,170,85,170,8
5,170
```

PROGRAMA 2

```
  10 READ  A$
  20 FOR A=0 TO 16 STEP 16
  30 LET D=1
  40 FOR X=1+A TO 15+A STEP 4
  50 FOR N=1 TO 17 STEP 2
  60 LET I=VAL A$(D): LET P=VAL
A$(D+1)
  70 PRINT AT N,X; BRIGHT INT (A
/16); INK I; PAPER P;CHR$ 32+CHR
$ 144
  80 PRINT AT N+1,X;P;I
  90 LET D=D+2
 100 NEXT N: NEXT X: NEXT A
 110 DATA "000102030405060711121
31415161722232425262733343536374
4454647555657666777"
 120 PLOT 128,0: DRAW 0,175
 130 PRINT AT 20,4;"BRILLO 0";AT
20,20;"BRILLO 1"
```

A LADRON, LADRON Y MEDIO

Joaquín Mateos Lagos nos ha escrito para decir que el truco «Las siete llaves», publicado en el número 16 de nuestra revista, es fácilmente soslayable. La solución que nos da es la de cargar el programa que ha sido salvado en forma de CODE, de esta forma:

CLERA 28999 : LOAD «nombre» CODE 29OOO una vez transcurridos unos 15 segundos de la carga, interrumpirla pulsando BREAK y teclear el siguiente programa:

```
10 FOR n=29000 TO 30000
20 IF PEEK n>31 THEN PRINT CHR
$ (PEEK n);
30 IF PEEK n<32 THEN PRINT " "
;
40 NEXT n
```

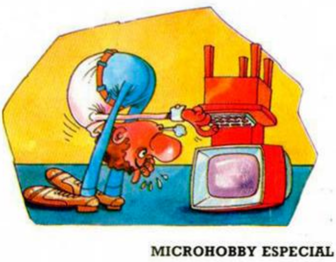
Esto permitirá ver claramente cual es la clave que debe ser introducida. A continuación, simplemente cargar bien el programa y, cuando se autoejecute y pida la clave, darle la correcta que hemos anotado anteriormente.

```
 100>FOR N=50000 TO 50053: READ
A: POKE N,A: NEXT N
 110 RANDOMIZE USR 50000
 120 DATA 33,0,64,17,31,64,6,192
,197,213,229,6,15,126,205,122,19
5,79,26,205,122,195,119,121,18,3
5,27,16,240,225,1,32,0,9,209,235
,9,235,193,16,223,201,197,6,8,79
,203,41,203,23,16,250,193,201
```

INVERTIR LA PANTALLA

Una vez más, nos hemos visto obligados a recurrir al código máquina para presentar una utilidad que, sin llegar a ser un largo programa, es bastante más que un truco; así que, como es corto, se puede introducir en cualquier subrutina de su propia aplicación siempre que corra en un Spectrum de 48 K (los poseedores de un 16 K, tendrán que desensamblar el programita y reubicarlo en otras direcciones de memoria).

¿Que qué es lo que hace el programa? pues invierte la pantalla, esto es, nos hace observar todo lo que esté dibujado en ella como si lo miráramos desde dentro del propio televisor; posibles mejoras a esta pequeña subrutina serían aumentar un poco su velocidad, aunque la que posee ahora es bastante aceptable, e invertir también los atributos de la misma forma que hacemos con los caracteres de pantalla.



MICRO DEALER
AMSTRAD CENTER

(Metro Goya, salida Felipe II)

Duque de Sesto, 52
28009 Madrid
Tel. 233 07 81

QuickShot™


+ INTERFACE: 3.950


+ INTERFACE: 4.350

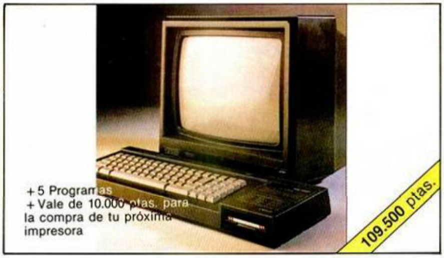

+ INTERFACE: 3.350

GARANTIA OFICIAL DE INVESTRONICA Y AMSTRAD ESPAÑA









Aceptamos tu Spectrum (descontándote 18.000 ptas.), al comprar tu nuevo QL con software en castellano (Garantía Investrónica) y además te regalamos cuatro superprogramas y 4 cartuchos vírgenes.

OFERTA CAMBIO

* Al comprar tu nuevo QL con software en castellano, cuatro superprogramas, 4 cartuchos vírgenes, garantía INVESTRONICA, aceptamos tu SPECTRUM, descontándote 18.000 ptas.

Te presentamos tu nueva Boutique de Microinformática. En ella vas a encontrar todo lo necesario para tu ordenador y por oferta de inauguración (válida hasta 15-12-85) *todo,* absolutamente todo, lleva regalo. ¡¡Compruébalo!!

los Supercincuenta

El mercado del software para Spectrum ha evolucionado muy rápidamente en un corto periodo de tiempo, durante el cual hemos podido comtemplar atónitos como se inundaba el mercado de productos y más productos, que en muchas ocasiones han creado cierta confusión entre los usuarios.
Nosotros, que hemos tratado de informar puntualmente de cuantas novedades iban apareciendo en el mercado, creemos que ha llegado el momento de hacer balance. Y por eso os ofrecemos a continuación una lista con los cincuenta programas que hemos considerado los mejores y que son ya imprescindibles en una buena programoteca.



HIGHWAY ENCOUNTER
Vortex
Arcade

Es la última creación de la compañía Vortex y nos introduce en una divertida aventura con un altísimo nivel de dificultad. El objetivo es dirigir a un pequeño robot y su corte de amigos hasta la base enemiga alienígena, por una carretera plagada de adversarios. Una vez allí hay que activar una potente bomba con la que pondremos fin a la comprometida situación.
El juego es en tres dimensiones y los gráficos son muy buenos.

GREMLIMS
Adventure International
Aventura gráfica

Otra reproducción de una película. Al igual que en todas las videoaventuras, en ésta hay que decir al juego en todo momento lo que debe hacer, pero con la salvedad de que esta vez todos los textos están en castellano, lo que es de agradecer. El juego sigue una lógica parecida a la de la película, con escenas muy similares. En definitiva, cualquiera que haya visto la película puede empezar a jugar sin problemas.





DYNAMITE DAN
Mirrorsoft
Arcade

Basado en Manic Miner, se trata de un magnífico programa de arcade con multitud de pantallas, todas ellas llenas de una desbordante imaginación.
La misión es ayudar al agente secreto Dan a encontrar la guarida del malvado Doctor Blitzen que junto con su linda asistente Donna, ha trazado un plan para crear el megarayo, un arma que en sus manos puede ser aniquiladora.



KUNG-FU
Buga-byte
Deportes

Como su nombre indica, el juego consiste en un cambate de Kung-Fu entre dos adversarios, en un «Dojo», una especie de gimnasio. Un detalle a tener muy en cuenta por lo bien conseguido que está, es el movimiento, ya que se pueden realizar movimientos conocidos de este arte marcial. La «Kata» o demostración, permite apreciar estos golpes aunque se puede suprimir si ya se conoce el juego. Hay tres intentos para llegar al máximo y la posibilidad de ver repetidos los golpes.



ABU SIMBEL PROFANATION
Dinamic
Videoaventura

Es la tercera parte de la serie empezada con Saimazoom y seguida por Babaliba. Esta vez el personaje está poseído por la maldición de Abu Simbel y debe ir al templo para recuperar su personalidad. Tendrá que sobrepasar 46 habitaciones en forma de laberinto si quieres llegar hasta el final. En ellas encontrará todo tipo de dificultades y trampas, algunas de ellas mortales. Tendrá que recoger amuletos para abrir determinadas puertas. Un diamante que recoja será decisivo al final, según su color, mientras que transbordadores le trasladarán a otras habitaciones.



ROCKY
Dinamic
Deportivo

Utiliza el estilo ya usado en las máquinas recreativas, el «Punch Out». El usuario maneja el boxeador que le da la espalda, mientras que el adversario se sitúa enfrente y un poco más arriba para dar una buena sensación de perspectiva. El objetivo es ganar a los cuatro contrincantes que son pesos pluma, medio por dos veces y pesado respectivamente, para de esta forma pasar a ser campeón mundial. Todo esto pasa por cinco fases del juego, cada vez más difíciles, hasta llegar a la quinta, que es el título mundial. Los datos y la energía de los contrincantes se encuentra en la parte inferior de la pantalla.



TIR NA NOG
Gargoyle games

Es un juego en el que para llegar al objetivo final es necesario tener una estrategia preestablecida ya que será muy necesaria según se vaya desarrollando El héroe es Cuchulain y el juego se desarrolla en Tir Na Nog, donde tiene que encontrar los fragmentos del sello de Calum.
La imagen simula ser una cámara de cine que sigue al héroe según la posición del usuario. Cuchulain deberá luchar contra sus enemigos con objetos que recoja mientras que en todo momento estará recibiendo mensajes de información y conocerá su posición, lo que es una gran ayuda al igual que la gran cantidad de movimientos que puede realizar.



GHOSTBUSTERS
Activision
Arcade

Los fantasmas intentarán destruir la ciudad con su energía, lo que tratará de ser evitado por los cazafantasmas el igual que la película del mismo nombre. El guardián de la puerta y el señor de la llave serán sumamente peligrosos si se encuentran en el templo de Zuul, centro de reunión de los fantasmas. Teniendo en cuenta la economía hay que elegir un vehículo para el transporte y hacer diversas compras de material como son el motorizado, el de captura y el de almacenaje, sin olvidarnos del importantísimo Detector de Energía PK o el intensificador de Imagen. A través de un mapa serán localizados los fantasmas para destruirlos, o al menos intentarlo.

KNIGHT LORE
Ultimate
Videoaventura

Un juego revolucionario en todos los sentidos con un nuevo estilo de programación denominado «FILMATION», lo que le da más libertad de desarrollo. El personaje es el mismo que en Sabre Wulf, pero mediante una transformación muy bien conseguida, al llegar la noche se convierte en hombre lobo. Hay que sobrevivir cuarenta días y cuarenta noches en un peligroso castillo con el fin de encontrar una pócima celosamente guardada por el mago y así acabar con la maldición. Para ello tendrá que enfrentarse contra todos los peligros que hay en el castillo de las mil trampas y llegar a la prueba final.





TED EL TECNICO
Hewson Consultants
Arcade

Muy parecido a Jet Set Willy. Ted debe ir todos los días a las veintiuna tareas que tiene asignadas. Pero aparte de la dificultad de no saber cuáles son ni dónde están, está la dificultad de las mismas. A veces será necesaria la ayuda de un amigo para realizar las tareas, ninguna imposible, y el tiempo será un factor determinante.

MAZIACS
DK'Tronics
Arcade

La misión del jugador es la de encontrar un tesoro en un gigantesco laberinto con el fin de llevarlo a una habitación destinada a preservarlo. Para conseguirlo hay que dirigir a un personaje armado con una espada. Unos seres muy extraños, los Maziacs, intentarán a toda costa

los Supercincuenta

evitar que el personaje cumpla su misión. Este puede luchar conta ellos siempre y cuando disponga de la espada, ya que después de cada enfrentamiento con los Maziacs la perderá y tendrá que coger otra de las varias distribuidas por el laberinto. Igualmente, hay repartidos platos de comida con los que se tendrá que alimentar si quiere conseguir su objetivo. Dispone de grandes ayudas para llegar al fin de su objetivo, como son, por ejemplo, los prisioneros que se encuentre por el laberinto.



SUPERTEST
Ocean
Deportivo

El objetivo consiste en superar ocho pruebas en dos días. Las pruebas a realizar son: PUNTERIA, con una pistola habrá que conseguir la mayor puntuación posible en dianas móviles; CICLISMO, hay que hacer un recorrido en el menor tiempo posible; SALTO DE TRAMPOLIN, hay que descender evitando a toda costa los palos; PIRAGUA, similar al ciclismo pero con un adversario; PENALTIS,



tratar de conseguir el mayor número de goles; SALTO DE ESQUI, teniendo cuidado en el contacto con el suelo, hay que saltar lo más posible; CUERDA, hay que medir la fuerza con adversarios cada vez más difíciles.

NODES OF YESOD
Odín
Videoaventuras

Charlie debe bajar a la Luna para encontrar un monolito de gran importancia, para lo cual debe hacerse con ocho llaves y cumplir su misión si quiere regresar a la Tierra. Tendrá que introducirse por los cráteres en un laberinto donde unos topos devoran los muros de las cavernas, lo que le dificultará su misión aunque le servirán



de ayuda si los atrapa. Charlie desafiará las leyes de la gravedad al estar en la luna, pero una caída desde gran altura puede traer malas consecuencias.

MATCH DAY
Ocean
Deportivo

Es el sucesor del conocido World Cup, en versión para Spectrum. Está realizado en tres dimensiones. Existe la posibilidad de elegir el color de los jugadores y el del campo. Con el efecto



tridimensional se puede conseguir darle al balón de cabeza o a ras del suelo. El balón puede incluso rebotar en los postes. El portero se puede mover. En definitiva, ha conseguido superar a su antecesor.

DEUX ES MACHINA
Autómata
Aventura Arcade

Un juego totalmente diferente. Más que un juego es una experiencia para ordenador. Traslada la historia al futuro, donde una computadora es dueña de todo. Pero un ratón logra entrar en el computador, lo que llevará consigo un proceso de reproducción humana. El usuario es el único responsable de esta formación, siendo el objetivo principal alcanzar la máxima perfección posible al desarrollar el DNA, el banco de memoria, los bancos de belleza y



atractivo, el proceso de fecundación del óvulo... A partir de aquí, la policía interna del computador, intentará eliminarlo. En la otra cara se convertirá en soldado y se hará viejo. Al final se obtendrá el porcentaje conseguido sobre el computador, lo que dará la victoria o el fracaso.



NIGHTSHADE
Ultimate
Videoaventura

En la presentación se puede ver una casa medieval en la larga noche de las sombras. El personaje se encuentra rodeado de figuras fantasmagóricas, y su misión es destruir a los cuatro enemigos de la ciudad, lo que no será nada fácil ya que cada uno sólo se podrá eliminar con su arma determinada que parpadeará al encontrase cerca del enemigo. Se puede disparar contra las figuras fantasmagóricas que hay por la ciudad con armas de fuego, pero no contra los cuatro enemigos. si se logra el objetivo final, volverá a lucir el sol en la ciudad de las sombras.

FRANKIE GOES TO HOLLYWOOD
Ocean
Videoaventura

Mezcla de sueños y realidad sin saber a ciencia cierta cómo distingirlos. Los objetos y los lugares por los que pasar son de lo más normales en apariencia, pero más adelante vendrá la ficción. La personalidad del personaje empieza siendo nula, por lo que habrá que conseguir un buen equilibrio. La acción se desarrolla en un barrio que parece muy tranquilo y el personaje tendrá que recoger varios objetos y curiosearlo todo. Se cometerá un asesinato y con 23 pistas habrá que encontrar al asesino. En algunas fases del juego será absolutamente necesario tener el nivel de personalidad al máximo.



DAMBUSTERS
U.S.Gold
Simulador de juego (combate)

Es la versión para el Spectrum creada por David Anderson e Ian Morrison acompañados por Marc



Craig. La acción se desarrolla en 1943, cuando el comandante Gibson debe realizar una peligrosa misión en la II Guerra Mundial al mando de setecientos hombres, la de destruir. objetivos enemigos, como por ejemplo, bases de vital importancia, en su Lancaster MK III. Hay tres niveles de dificultad y tres posibilidades de juego: una de práctica, otra que empieza en pleno vuelo y otra que empieza segundos antes del despegue.

THE ROCKY HORROR SHOW
CRL
Videoaventura

La acción se desarrolla en un antiguo teatro decorado con rayos de tormenta. En general, el ambiente está



muy conseguido. El teatro tiene varios pisos y la misión es recoger las quince partes de un cuadro y llevarlas hasta el lugar correspondiente. En el escenario está prisionero-a el compañero del protagonista o de la protagonista, según se elija. Cada puerta se abre con su llave respectiva que habrá sido recogida antes. Para pasar por el gimnasio hay que desactivar el rayo energético. Por todas partes aparecerá un punkie, un motorista, una dama de nagro y algún personaje más de los que habrá que huir.



MATCH POINT
Psion
Simulador deportivo

Se trata de un partido de tenis muy bien reproducido Al principio del juego hay un menú que nos permite decidir contra quién jugamos, la duración del partido y la modalidad de movimiento. Respecto a los gráficos son muy buenos ya que encontramos detalles como la sombra de la pelota. En cuanto a los jugadores, disponen de cuatro movimientos aparte de golpear la pelota (de drivi o de revés) con una gran sensación de perspectiva. Los niveles de dificultad son tres: cuartos de final, semifinal y final, aunque este último muy difícil de superar. En definitiva, uno de los mejores juegos del mercado.

HERBERT'S DUMMY RUN
Mikro Gen
Videoaventura

Herbert, el personaje principal, no es otro sino el bebé de Everyone's a Wally. Herbert se ha perdido en unos grandes almacenes y la misión es llevarlo a su casa en un tiempo real de cuatro horas y media, superando las fases del juego. Se necesitarán diversos objetos para afrontar las múltiples fases que estarán relacionadas con el final del juego. La medida de energía de Herbert está situada en la parte inferior de la pantalla y podrá ser recuperada tomando golosinas que hay por los almacenes.



THE WAY OF THE EXPLODING FIST
Melbourne House
Karate

Un magnífico juego empezando por los decorados y siguiendo por los soberbios movimientos que convertirán al juego en un auténtico combate una



vez que se aprende a dominar los dieciseis movimientos de que se dispone entre defensivos y ofensivos. Para aprender se puede elegir la opción de dos jugadores y manejar uno solo. El decorado irá cambiando a medida que se vaya derrotando al contrario, lo cual depende de la belleza del movimiento con que se consiga. Cada combate lo gana el mejor a tres puntos o el que mayor puntuación tenga al finalizar el tiempo de cada uno de ellos.

SPY HUNTER
U.S. GOLD
Arcade

El objetivo no es hacer más y más kilómetros, sino llegar al final del juego. Un agente ha de cumplir una misión determinada, pero para ello tendrá que evitar a diversos enemigos, el Señor de la carretera, el Navajas o el Pistolas, que tratarán por todos los medios de impedir el éxito de la misión. Pero el peor de todos los enemigos es un implacable helicóptero que perseguirá al agente.
Habrá veces que tendrá que atravesar un río en barco si el puente está destruido y superar todas las dificultades, aunque también recibirá colaboraciones de aliados.





INTERNATIONAL BASKETBALL
Imagine
Deportes

Un magnífico juego de simulación de Baloncesto en el que se han cuidado al máximo todos los detalles para conseguir una buena ambientación.
Tiene seis niveles de dificultad y nos permite elegir el color del campo y los personajes.
La técnica utilizada es parecida a la del Match Point, nosotros controlamos a un solo jugador, el que lleva la pelota y el que marca el ritmo del resto del equipo.
Es la última producción de Imagine y uno de sus mejores juegos.



HYPERSPORTS
Imagine
Deportivo

Es la versión para el Spectrum del conocido juego de las máquinas. Hay que ir superando pruebas cada vez más difíciles: natación, tiro con arco, plato, triple salto y levantamiento de pesas. Cada cierto tiempo habrá que pulsar la tecla del aire, siendo muy importante hacerlo en el momento preciso. Para superar una prueba y pasar a la siguiente hay que rebasar un baremo de dificultad. Una vez conseguido se vuelve al principio con mayor dificultad.

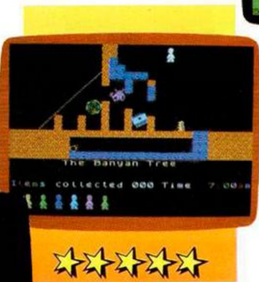


los Supercincuenta

KOKOTONI WILF
Elite
Arcade

Es un duelo de habilidad y reflejos contra el ordenador. Kokotoni Wilf deberá recoger un determinado número de llaves desde la era primitiva hasta la actualidad que han sido escondidas por el mago «ULRICH», su maestro. Tendrá que enfrentarse contra todo tipo de peligros típicos de cada época. Para pasar de una a otra debe coger las llaves correctas aumentando la dificultad a medida que avanza el juego. Sólo hay que manejar tres controles: izquierda, derecha y volar. Un detalle a agradecer es que si falla, se vuelve al mismo punto y no al principio.







CONDENAME MAT
Micromega

Los Myons atacan la tierra y ésta ha de ser defendida mediante una revolucionaria nave espacial con un gran número de mandos e información a controlar. Está formado por tres niveles de dificultad: el de práctica, el de invasión moderada y el de invasión absoluta. Aparte, el usuario puede ser comandante o simplemente piloto tratando a toda costa de evitar la invasión de los cazas, crucero y estrella base de los Myons. El dominio absoluto de los mandos es imprescindible para defenderse. El manual de instrucciones es muy completo.

JET SET WILLY
Software Projects
Arcade

Es la continuación de Manic Miner. Esta vez el minero es rico y vive en una gran mansión. Tras una ruidosa fiesta intentará entrar en su habitación para dormir, pero su ama de llaves no le dejará hasta que recoja todos los desperdicios de la fiesta. Para hacerlo debe estudiar bien todos sus movimientos porque la casa está llena de lugares raros que le dificultarán su misión. Respecto a la mecánica, es muy parecida a la de Manic Miner pero ligeramente superada. Consta de muchas pantallas comunicadas entre sí.

DECATHLON
Ocean
Deportivo

Basado en la prueba deportiva del Decathlon, y supervisado por Daley Thompson's, actual campeón de esta disciplina. Hay que desarrollar las 10 pruebas que la componen: 100 m. lisos, salto de longitud, lanzamiento de peso, salto de altura, 400 m. lisos, 400 m. vallas, salto con pértiga, lanzamiento de martillo, lanzamiento de jabalina y 1500 m. lisos. Para ello se cuenta con tres atletas que irán desapareciendo si no superan las pruebas de clasificación. Un factor muy importante para superarlas es coordinar la relación coordinar la relación ángulo-velocidad.



TUTE
Investrónica
Cartas

Una auténtica partida de cartas (poco frecuente en los juegos) contra el ordenador. Nuestras cartas son visibles en todo momento mientras que, lógicamente, las del resto de jugadores están tapadas. Las reglas son las mismas del juego, es decir, se puede cantar y hacer renunci[illegible], aunque en ese caso el ordenador nos lo dirá enseguida. Nuestro compañero está situado enfrente de nosotros. Al final de cada mano, el ordenador hará el recuento de todos los tantos, incluidos los cantes y el monte. Ideal para pasar un rato entretenido.

GIFT FROM THE GODS
Ocean
Videoaventura

El héroe de este juego, Orestes, ha vuelto al palacio de Mycenae para recuperar su reino, vengar a Agamenon, su padre, y rescatar a Electra, su hermana. El palacio forma un laberinto con cámaras conectadas entre sí y con objetos de forma especial denominadas formas euclídeas. Ayudado por los dioses, Orestes tendrá que hallar seis objetos y colocarlos correctamente en la cámara del guardián para conseguir el éxito. Criaturas ilusorias tratarán de impedírselo, pero con la ayuda de Electra, de una espada regalo de Zeus y las siete Lágrimas de Icaro, conseguirá su objetivo final.





CYCLONE
Vortex
Arcade

Está basado en TLL aunque es mucho más original. La idea principal es la del salvamento de los habitantes de unas islas con un helicóptero. Para ello hay que partir de la isla base con el depósito lleno. En la parte derecha de la pantalla aparecerán los mandos. Un ciclón será un continuo peligro para el aparato, pero con un mapa, es localizable en cualquier momento. Con una cuerda se rescatará a los necesitados y en caso de apuro se puede aterrizar en determinadas zonas teniendo mucho cuidado.



AVALON
Hewson Consultants
Aventura gráfica

Maroc (el mago), utilizando sus poderes y hechizos según su conveniencia, debe atravesar más de 200 habitaciones, túneles y cuevas a través de ocho niveles distintos co[illegible] fin de destruir al Señor del Caos. Aparte de sus hechizos, encontrará objetos que pueden serle útiles. El recorrido es muy variado, ya que irá pasando por sitios como barracones, cuevas naturales, laberintos, catacumbas... Dependiendo del nivel de dificultad, cada hechizo servirá para una ocasión determinada. Es un tipo de programación ligeramente distinto a los actuales, lo que le hace bastante atractivo.

los Supercincuenta

EVERYONE'S A WALLY
Mikro-Gen
Videoaventuras

Wally, personaje ya famoso gracias a Pyjamarama, aparece acompañado esta vez de Wilma (su mujer), Tom (el «punkie»), Dink (el fontanero), Harry (el «hippie» electricista) y Herver (el bebé), cada uno con su personalidad propia fundamental en el desarrollo del juego aunque sólo se puede manejar a Wally. El objetivo es abrir la caja fuerte del banco, cuya combinación está esparcida por toda la ciudad, haciendo mientras tanto las labores cotidianas, lo que nunca es sencillo.



AIR WOLF
Magic-Team
Arcade

La misión consiste en rescatar a cinco científicos cautivos en una base subterránea del desierto. Habra que ir atravasando cuevas que se podrán destruir pero que se volverán a formar al cabo del tiempo. Después de las dos primeras pantallas comienza el descenso con muchos peligros que lo dificultarán, aparte de los numerosos túneles y pasadizos que convierten a este juego en uno de los más difíciles del mercado.



SKOOL DAZE
Microsphere
Videoaventura

Es uno de los juegos más originales del mercado. La acción se desarrolla en un colegio donde impera la más absoluta anarquía. Para evitar que Eric, el protagonista, sea expulsado del colegio habrá que llegar hasta la caja fuerte del director y sacar su expediente. La asistencia a clase es obligatoria ya que si no podría ser expulsado del colegio. La rivalidad entre alumnos es continua, pero siempre trantando de no ser vistos para no ser castigados. Los movimientos son muchos y muy variados. Es una manera diferente de asistir a clase.





TRASHMAN
New generation
Arcade

Mediante la habilidad y la rapidez, el objetivo principal es recoger las basuras de un barrio evitando en lo posible los peligros que acechan. Hay que tener cuidado al cruzar la calle ya que podría ser atropellado; o no pisar el césped debido al perro que lo guarda; o no beber mucho en los bares para rendir más. Por tanto, hay que recoger basuras en un barrio muy bien ambientado y volcarlas en un camión en un tiempo limitado. No es nada fácil.

BANDERA A CUADROS

Psion

Arcade

El objetivo es el de pilotar un coche de carreras por un circuito previamente elegido, entre diez que se presentan, en el menor tiempo posible. Para ello, aparte de la elección del circuito, hay que elegir a nuestro gusto un coche entre tres opciones. Antes de cruzar la línea de meta habrá varias dificultades de distinto nivel como son los cristales, piedras, aceite... Los efectos son muy buenos ya que algunas veces habrá que entrar en boxes o, si se pincha, los bordes de la carretera vibrarán como si fuese en la realidad.

La sensación de realismo está bastante bien conseguida y parece en algunas ocasiones que estamos realmente frente al volante de un auténtico coche de fórmula 1.

El programa es de lo mejor del género y de alguna forma el que rompe un poco el molde. Es el primer programa de simulación deportiva propiamente dicho y supone un éxito muy importante para la compañía PSION.

PSYTRON

Beydon

Arcade

El enemigo tratará de destruir una base que es la que hay que proteger y que está compuesta de varias instalaciones, cada una con una misión específica y comunicadas entre sí. Todas ellas tienen su propio personal que irá siendo eliminado por el enemigo, mientras que para recuperarlo habrá que llevarlo a la unidad médica. Hay veces que, ante al continuo ataque del adversario, habrá que decidir qué elemento defensivo conviene proteger en detrimento de otro. Hay seis niveles de dificultad que sólo se podrán superar con un gran conocimiento del juego.

El programa tiene varias fases distintas y nuestra misión va cambiando a medida que vamos avanzando en alguna de ellas, siendo más difíciles cada vez.

Gráficamente es un programa de lo mejorcito que hay porque aunque está estructurado a base de grandes bloques de pantalla, éstos resultan muy atractivos para el jugador y también significativos a la hora de controlar el sistema de juego.

UNDERWURLDE

Ultimate

Videoaventura

El juego , la segunda parte de la trilogía de Sabreman, está basado en un complejo laberinto lleno de dificultades. Hay que llegar hasta el palacio de la noche, donde espera un diablo contra el que habrá que luchar. En el camino se encuentran brujas aladas, bolas de fuego, diamantes mágicos, sirenas en forma de plantas venenosas, cráteres volcánicos y un sinfín de inconvenientes más. Este juego continúa con la saga empezada con Sabre Wulf, superándolo incluso, ya que los detalles se hacen con un gran realismo y el nivel de dificultad es muy alto.





COMBAT LYNX

Durell

Simulación

Dentro del grado de dificultad que se escoja, hay que defender un ejército que se irá moviendo por un mapa en el que, a veces, se podrá ver simultáneamente el avance de los enemigos. Usando la estrategia hay que eliminarlos, bien en tierra o bien en aire. Se dispone de muchos datos y movimientos para conseguir el objetivo final como, por ejemplo, una micropantalla que nos comunica detalles de gran importancia, controles de vuelo o posibilidad de conseguir armamento, recoger personal, etc. En pocas palabras, hay que mezclar estratégicamente todos estos elementos como si fuera la realidad.

SABRE WULF
Ultimate
Arcade

La acción se desarrolla en una selva con forma de laberinto. La misión es recoger las cuatro partes de un tesoro y salir de la selva. Para aumentar la puntuación hay diversos objetivos que se pueden coger. El recorrido está salpicado de muchos alicientes que lo dificultarán como son enemigos que aparecen en cualquier momento (unos vulnerables con una espada y otros no), animales que nos atacan, negros que nos persiguen, templos, tesoros, lagos, plantas curativas, etc. Los gráficos en general son muy buenos.



HORMIGAS
Quicksilva
Arcade

Un juego presentado en tres dimensiones en el que el objetivo es rescatar a una dama (o caballero, a elección) que se encuentra en una ciudad invadida por las hormigas. Contra éstas se pueden usar bombas en caso de necesidad. Para todo ello disponemos de un tiempo limitado. Respecto a los gráficos, están muy bien conseguidos ya que la imagen se desplaza a medida que lo hace el

los Supercincuenta



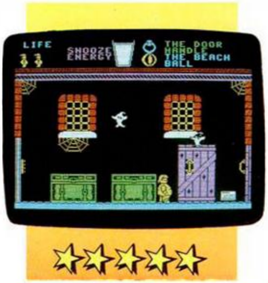
personaje; mediante una tecla se ve la escana desde ángulos distintos y, en caso de perderse el personaje, mediante otra tecla se verá su posición.

FIGHTER PILOT
Digital integration
Simulador de vuelo

El juego desarrolla una pantalla tridimensional desde la cabina de un avión (basado en el avión de combate F-15 Eagle USAF).



Para conseguir el objetivo hay que localizar el avión enemigo mediante mapas y rádares situados en la parte inferior de la pantalla, al igual que los mandos. Una vez hecho esto, habrá que destruirlo con las sofisticadas armas de que se dispone a la vez que habrá que estar atentos a los mandos del avión que se pilota.



PYJAMARAMA
Mikro-gen
Arcade

Es una representación de un sueño. El personaje, que no es otro sino el que está dormido, debe conseguir poner en funcionamiento un despertador para que acaben sus pesadillas. Para hacerlo debe atravesar pasillos y habitaciones, pero con muchos problemas como objetos que atacan, dardos asesinos, habitaciones que se mueven..., es decir, como si de una pesadilla real se tratara. Por el camino se pueden recoger objetos con el aliciente de que no se sabe se servirán o no. Las sorpresas no se acaban nunca.

TLL
Vortex
Arcade

La misión consiste en destruir objetivos enemigos situados en una ciudad. Para ello se dispone de los mandos de un avión con cuatro tipos de movimientos, aparte del despegue y aterrizaje (éste es muy importante). Para facilitar la destrucción del enemigo hay un mapa el que se accede pulsando una tecla y que nos indica su posición. Entre la información de la que se dispone hay un rádar de corto alcance, un altímetro, indicador de gasolina, blancos enemigos y el mapa. El manejo del avión es bastante sencillo.



MANIC MINER
Bug-Byte
Arcade

Disponemos de un personaje con tres vidas, el minero Willy. Hay que recorrer veinte cuevas

¡FANTASTICO!



Si quieres un ordenador de «una pieza» piensa en el AMSTRAD CPC 464. Tendrás un ordenador de una vez por todas. Gracias a sus 64K RAM y 32K ROM y a sus casi ilimitadas posibilidades de crecimiento, tienes garantizado que el ordenador CPC 464 no se te quedará pequeño.

COMPLETO

Además, gracias a su monitor (color o fósforo verde) de alta resolución (hasta 640 x 200 pixels direccionados individualmente) y a su unidad de cassette incorporada al teclado, podrás disfrutar de tu AMSTRAD de una manera independiente, prescindiendo del televisor y del radiocassette de tu casa (a veces tan solicitados).

¿Y QUE ME DICES DE LOS PROGRAMAS?

Actualmente ya hay cientos de ellos disponibles en España. Sin olvidar que son varias las revistas dedicadas sólo a AMSTRAD y que el número de libros y periféricos del CPC 464 crecen día a día, potenciando así la creatividad de tu ordenador personal.

CARACTERISTICAS TECNICAS

- Microprocesador Z80 (4MHz).
- Memoria de 64K RAM y 32K ROM.
- Gráficos de alta resolución de hasta 640 por 200 pixels direccionables individualmente.
- Unidad de cassette incorporada en el teclado.
- Monitor color o fósforo verde incluido en el Sistema.
- Texto en pantalla de 20,40 y 80 columnas por 25 líneas.
- LOCOMOTIVE BASIC ampliado.
- Paleta de 27 colores y efectos de «flash».
- Teclado profesional tipo QWERTY con bloque numérico y teclas para cursor independientes.
- Salida Centronics paralelo.
- Lector de discos de 3" (180K por cara) opcional (con CP/M y Dr. LOGO incluidos junto a la unidad de disco).
- Manuales en castellano.

Al comprar tu ordenador CPC 464, AMSTRAD ESPAÑA te obsequia con 8 cassettes de programas y el libro «Guía de Referencia BASIC para el programador».

Exije la **GARANTIA AMSTRAD ESPAÑA UNICA VALIDA PARA ACCEDER AL SERVICIO TECNICO OFICIAL.**

PRECIO:

— 66.900 ptas.
(monitor fósforo verde)

— 95.900 ptas.
(monitor color)

¡¡Increible!!

AMSTRAD ESPAÑA

GARANTIA INDESCOMP

Avd. del Mediterráneo, 9 - 28007 Madrid Tels. 433 45 48 - 433 48 76 - Telex 47660 FAX - 4332450

los Supercincuenta

diferentes evitando caerse al vacío, los desprendimientos y eludiendo a los monstruos que la habitan. Todo ello crea un nivel de dificultad bastante alto. Es un juego que está muy bien conseguido en todos los aspectos y desarrollado con una gran imaginación. Para completar todo esto, el sonido y la música son también muy buenos. En definitiva, es un gran juego.



ALCHEMIST
Imagine
Aventura gráfica

El alquimista es el único ser sobre la tierra capaz de derrotar a Warloc dentro del castillo maldito lleno de trampas. Para ello debe encontrar las 4 partes de un conjuro a la vez que objetos que pueden serle útiles. La stamina será la que mida la cantidad de energía que se pierde luchando contra los guardianes y se recupera alimentándose por el camino. Cada vez que recoja un conjuro debe llevarlo hasta su santuario y guardarlo en su cofre, labor que será más fácil si se



convierte en águila y vuela. Una vez hallados los cuatro conjuros, estará dispuesto para enfrentarse a Warloc.

FULL THROTTLE
Micromega
Arcade

El riesgo y la velocidad van juntos en este juego en tres dimensiones. Antes de empezar hay que elegir varias cosas entre diversas opciones como, por ejemplo, un circuito entre los diez mejores del mundo (incluido el Jarama).



Siempre, el piloto que hay que dirigir parte en última fila de una parrilla de cuarenta corredores y debe esquivar a todos ellos para llegar el primero a la meta, lo que no va a ser nada fácil. La inclinación en las curvas y los efectos sonoros del motor y el derrapaje son muy buenos.



BABALIBA
Dinamic
Videoaventura

Es la segunda parte del programa Saimazoom. Johny Jones se encuentra esta vez en el palacio de Hosmimumarack, donde tiene que apoderarse del tesoro oculto, raptar a la princesa y rescatar a un compañero de las mazmorras. Hecho todo esto debe volver a la salida y juntar las palabras (en el mismo orden) que forman BABALIBA. Por supuesto que no será un camino de rosas ya que los enemigos acechan por todos lados y el palacio consta de cuatro plantas. El juego tiene 135 pantallas.

¡EXCLUSIVA MUNDIAL!

Editado por: SINTAX, S.A.

YOUR COMPUTER

La Revista de ordenadores de mayor venta en toda Europa

¡SE PUBLICA DESDE AHORA EN ESPAÑA, EN FORMA DE CASSETTE!

Sí, ya está confirmada la sensacional noticia. Muy pronto estará ***en los quioscos*** *de toda España una selección de los mejores juegos y utilidades publicados por la prestigiosa Revista británica «YOUR COMPUTER», editados en cassette de alta calidad y con instrucciones en castellano.*

El ***prestigio*** *alcanzado por Your Computer, tanto en Inglaterra como en España y otros países, se debe, de una forma muy especial, a la* ***gran calidad de los programas*** *que publica, la mayor parte de ellos en Código Máquina, y con la utilización de rutinas y técnicas de programación muy depuradas.*

Ahora, a un precio inmejorable, podéis tener acceso a estos programas, ***evitandoos*** *la difícil tarea de* ***teclearlos*** *en vuestro ordenador.*

¡Y ***cada mes*** *estará en la calle una nueva cinta!*

Si no encuentras la cassette de **«Your Computer»** en tu quiosco o tienda de informática, solicítala a nuestras oficinas:

SINTAX, S. A.
«YOUR COMPUTER»
Paseo de la Castellana, 268
28046 Madrid

Envía tus señas completas, teléfono **y marca de ordenador** e incluye **talón bancario,** o remite **Giro Postal** por el importe.

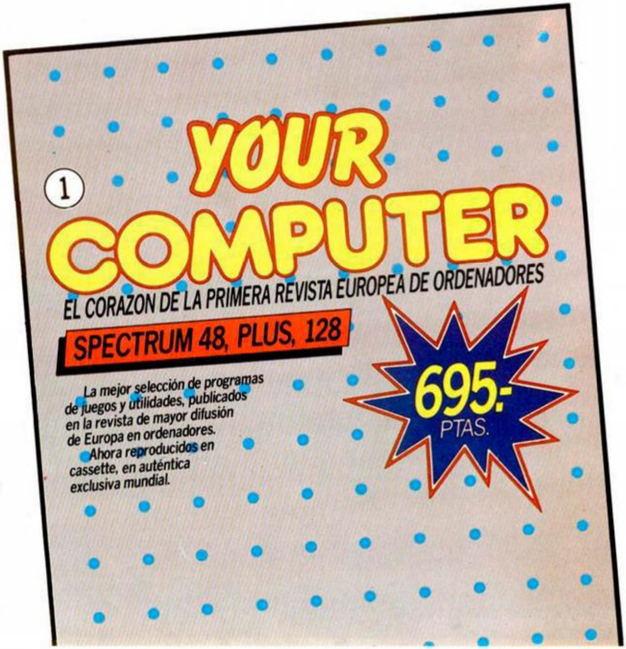
No te cobraremos gastos por el envío.

Si prefieres pagar **contra reembolso,** entonces incluye, junto a tu pedido, dos sellos de 50 ptas. cada uno para gastos de envío.

TAMBIEN DISPONIBLE PARA

COMMODORE 64 y

AMSTRAD



Colega elefante

por Jesús Torbado

Tal vez lo único que me da miedo es el frío, porque sé que estoy condenado al frío, como todos los de mi raza. He estado a punto de disolverme en esas terrible tardes de calor, trabajando ciegamente durante horas, sudando por todos los rincones de la piel; y El ha tenido que comprender tanto esfuerzo y me daba algún respiro. Sin embargo, no sentí miedo, no el miedo como ahora. Se me nublaba el cerebro, transpiraba hasta en el interior del corazón, pero en ningún momento pude advertir este pánico que me agita ahora, el pánico a la desaparición definitiva, al frío. Desde que habló del asunto con su abuela y me informó (supongo que parcialmente) de la actitud de ella, «su buena disposición inicial», dijo —pero una verdadera decisión, quizá—, del renacido entusiasmo que sentía ante el futuro, ese frío una vez remoto e imposible, el gran hielo que nunca pude imaginar en mi juventud, comenzó ya a fijarse en mis huesos.

Se sentó como siempre, aparentemente triste pero animoso, y me saludó con las mismas palabras de todas las tardes:

—Hola, Elefante— dijo.

—¿Eres tú?

—¿Acaso conoces a otro?

—Conozco a mucha gente— dije yo.

—Te daré mis coordenadas.

Fue marcando, compulsivo, la más secreta de sus claves, muy deprisa, según su costumbre; una clave que apenas utilizaba. Parecían sus dedos los de un pianista loco. Luego añadió:

—No me gusta esa infidelidad. Sólo me conoces a mí.

—Te conozco a tí, conozco a tus amigos... Conozco a tus amigas.

—Pero sólo trabajas para mí.

—Cierto.

—Veamos un proyecto que se me ha ocurrido.

Parecía poseído de la misma hermosa furia de los primeros tiempos, cuando llegué a su casa y me sorprendí al sentir sus dedos azotados por una tempestad. Imagino que sin tiempo para pensar en lo que de verdad quería, tal vez sin haber reflexionado lo suficiente, ideaba cifras y letras que me comunicaba a toda velocidad. Tampoco se mostraba particularmente respetuoso con las leyes de nuestro común idioma y debía rectificar continuamente. Yo sabía que me estaba tendiendo una trampa, que buscaba una razonable disculpa; si me confesaba derrotado en aquel momento, el frío que empezaba a rozarme los huesos terminaría paralizándome.

—Incorrecto. Incorrecto— decía yo.

—¡Está bien!

—Incorrecto. Repite, por favor.

—¿Qué pasa?— preguntó El.

Y yo repetí, algo asustado:

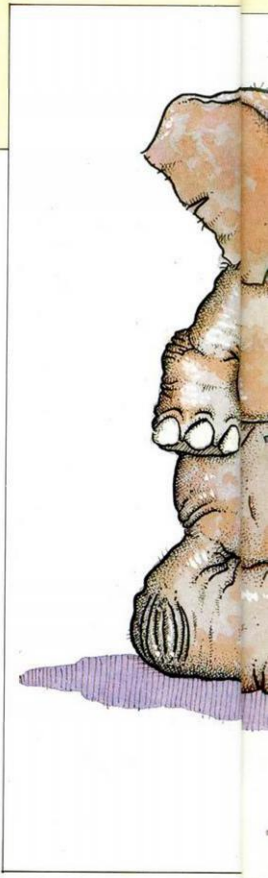
—¿Qué pasa? ¿Qué pasa?

De nuevo sus agitados dedos comenzaron a teclear enloquecidos símbolos y aunque nuestra comunicación era ya antigua y muy precisa, no encontraba dentro de mí una respuesta apropiada a aquellla furia.

—Incorrecto... Incorrecto...

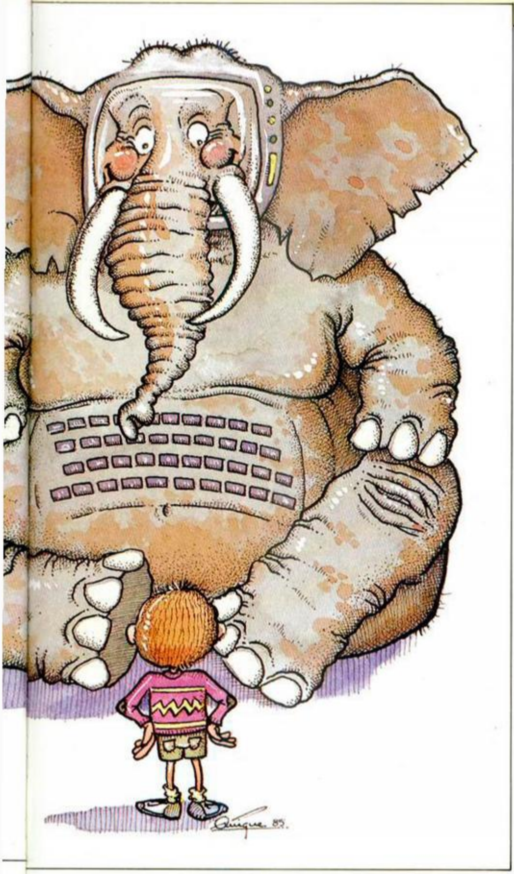
¿Qué otra respuesta podía darle? Advertía su creciente nerviosismo y no estaba preparado para defenderme de él, porque no me habían educado para ello. Sólo respuestas negativas, las cuales, por otro lado, le enfurecían más. ¿Cómo calmarle? Durante un par de minutos intentó aún transmitirme su confusión y su ira. Decidí entonces resolver mis propias dificultades anulando simple y radicalmente la comunicación. El verde ojo cuadrado se quedó mirándolo con fijeza; un levísimo parpadeo fluorescente le informaba de que yo seguía vivo, asustado pero vivo, y cerca de El, como siempre.

En realidad, me sentía más Colega que nunca e intentaba encontrar en nuestra historia común los apoyos que pudieran salvarme. En el camino de la emoción y el entusiasmo hasta la ira y la indiferencia, tenía que haber un puente que nos mantuviera unidos, el territorio eléctrico que durante tantas horas nos había conservado amigos y compañeros. ¿Sería exacto hablar de egoísmo en aquella investigación mía?



Al fin y al cabo, era mucho lo que yo le había dado. No sólo había permanecido durante casi dos años a su exclusivo servicio, como ayudante, confesor y cómplice, sino que me había esforzado por salvarle de la dramática y repentina soledad; no es inmodestia añadir que lo había conseguido.

Antes de conocerle, únicamente sentía interés por sus propias desdichas, como él había asegurado. Al parecer, en los primeros meses después de la desaparición de su padre, se había sentido más feliz



que nunca. Su madre y su abuela vivían exclusivamente pare él e incluso el padre le escribía y le telefoneaba de vez en cuando para interesarse por sus estudios, por su vida. La comunicación, que era incluso más intensa que cuando vivía a su lado, fue debilitándose poco a poco: cada vez más esporádica, más imprecisa, más amorfa. Su padre iba convirtiéndose en una sombra lejana de cuya visión estaba cada vez menos seguro, perdidos ya los matices del recuerdo.

Y tal lejanía coincidió casi con el súbito despego de su madre. Nunca lo hubiera creído posible, después de tantas lágrimas como ella derramó sobre su cabeza, después de tanto infinito amor expresado aburridamente a toda hora: la soledad compartida, el desprecio hacia el hombre que tanta desgracia les había traído, las diarias dificultades porque su dinero se retrasaba o se perdía. ¿Cómo un padre podía tener un sentido tan aleatorio de sus responsabilidades?, preguntaba ella.

Debería habérselo preguntado a sí misma un poco más tarde. Empezó saliendo mucho de casa y luego ya no entraba. El advenedizo era un tipo nueve años más joven que ella, muy elegante, farmacéutico de profesión. Conocía yo su historia quizás mejor que la de ningún otro miembro de la familia, porque a poco de haber aprendido nuestro sistema de comunicación iba ofreciéndome todos los datos que aprendía de él: los colores dominantes de sus corbatas, un cálculo de las ventas diarias en la farmacia, las horas aproximadas que pasaba con su madre y el número y duración de sus llamadas telefónicas. Fue quizás nuestro primer gran juego.

En realidad, se trataba de la cuantificación de su desdicha, si es que esto puede hacerse o decirse. Cuantos más datos pudieran acumularse sobre aquel intruso, mejor podía aborrecerle. Muchos de tales datos los había ido reuniendo antes de conocerme a mí, ya que inmediatamente puso manos a la tarea. La tragedia del doble vacío, el del padre y el de la madre después, lo había ido apartando también de sus amigos posibles, ante quienes no deseaba en absoluto hablar del asunto. La mayor parte de ellos vivía en familias oficialmente estables y sólidas, «mi padre va a comprarme una moto», decía uno, por ejemplo, «le pediré a mi madre dinero para el concierto», decía otra, y él carecía de puntos de contacto tan vulgares, así que fue recluyéndose en esta habitación que ahora compartimos, manoseando libros que era incapaz de leer, oyendo músicas que empezaban a asquearle, atendiendo sin ánimo a las palabras de desganado amor de su abuela, la única por lo demás que intentaba arrancarle de aquel pozo, sola también pero firme y decidida... Después, ella fue la que tuvo la idea. Ni su padre ni su madre pensaron en mí, sólo la abuela, guiada quizás por algún anuncio de la televisión, o por una palabra suya expresada sin convición excesiva. Lo estaba intentando todo para salvarlo.

De este modo fue como me condujeron a su casa el mismo día de su cumpleaños. Diecisiete, creo recordar.

—He tenido que empeñarme, pero te hará compañía— dijo la abuela mientras miraba mi cuadrado

—Hola, Elefante —dijo.
—¿Eres tú?
—¿Acaso conoces a otro?
—Conozco a mucha gente —dije yo.
—Te daré más coordenadas.

ojo ciego. —Me han dicho que es muy útil y entretenido. A ver si te gusta.

No quiero recordar las historias que me desfavorecen: el enfado de la madre cuando se enteró, la sonrisa imbécil del farmaceútico exquisito, que tenía planeado ya, según afirmó, instalar en su negocio un aparato veinte veces más poderoso que yo. Lo cierto es que El se lanzó como un atleta desesperado a aprender el nuevo lenguaje que necesita para conversar conmigo. No le llevó demasiado tiempo pues, aunque abúlico y desanimado en sus estudios hasta entonces, era un muchacho inteligente y esforzado en aquello que no significaba obligación o deber. Por otro lado, estaba magníficamente dotado. Está, quiero decir.

En apenas un verano acabó con todos los marcianos del universo y en récords inverosímiles se hizo campeón de todos los deportes, salvó los infinitos obstáculos que otros habían puesto en la creación



de sus juegos... Y muy pronto comenzamos a jugar con el farmaceútico, con la madre, con las probabilidades de que a su abuela se le quemase la comida. La mujer era muy olvidadiza y El pronto consiguió enseñarle un sistema elemental de recuerdo por el procedimiento simple de mirar mi ojo verde.

—Vamos a ver, Elefante— me decía, —has de meterte esto en la memoria.

Datos innumerables e inútiles con los que El lograba consolarme... Redujo toda la vida del colegio —casi mil alumnos— a un puñado de cintas sabiamente organizadas; al cabo de un año poseía un maravilloso tesoro de relaciones, secretos, adivinaciones. Aquello que para la mayoría había sido intranscendente o efímero, enfados súbitos, repentinos amores, ausencias, sospechas, diatribas, golpes, era trasvasado a mi memoria y cuidadosamente clasificado. Especialmente los datos referidos a sus mejores amigos y sus mejores amigas, todos los que habían ido volviendo a la sombra de su prodigioso conocimiento (que, naturalemnte, aplicaba también al campo docente, con investigaciones sobre probabilidades de preguntas en exámenes, previsión de calificaciones, tipología de los profes, etcétera), quedaban de tal modo organizados que sabía de sus compañeros incluso más que ellos mismos.

—Pero ¿cómo puedes acordarte?

—Me sopla mi colega— decía El. —Tiene memoria de Elefante.

Nadie había hecho trabajar tanto a un ordenador como El, nadie imaginaba siquiera que fuese posible un esfuerzo tan detallado, minucioso y perfecto. Y gracias a ese esfuerzo común, suyo y mío, alcanzó algunos éxitos destacados entre condiscípulos y profesores, pues algunos días vacacionales se veía invadida nuestra habitación por compañeros que querían divertirse con nuestros conocimientos. Sin falso orgullo, debo señalar que contribuí en buena medida a brindarle una victoria —sólo parcial, eso sí— con una muchacha llamada Azucena, la que pasaba por ser la más guapa de los cursos superiores... Claro que en ese terreno prefiero no meterme, aunque no lo he olvidado, no.

venturoso destino hasta aquel momento. Compañero fiel y voluntarioso, la mitad de su alma en tantas horas... Pero, lo mismo que las suyas, también mis fuerzas tienen un límite. Y si El tuvo miedo un día, también lo tengo yo hoy. La soledad es el frío y el frío es la muerte.

—Comunicación restablecida. Comunicación restablecida— conseguí decir con tremenda desgana y disimulando la herida de estas palabras con un brillante parpadeo multicolor de mi único ojo.

—¿Preparado?

—Preparado.

Mi amigo parecía haberse tranquilizado un poco. Ahora fue tecleando despacio, como si temiera equivocarse, con las pupilas clavadas en mí y una expresión concentrada y severa. Tenía a su lado varios manojos de papeles impresos y manuscritos, algunos muy lujosos, y de ellos entresacaba las claves que me iba proporcionando. Durante todo un curso había estado ahorrando y enviando cartas a su padre y mensajes directos a todos sus conocidos, el farmaceútico incluido —que llevaba camino de convertirse en su padrastro, según deduzco—... Tenía mucho dinero.

Y lo que me estaba pidiendo resultaba profundamente desolador. Yo almacenaba silencioso cuantos secretos me iba confiando. De cada uno de los ordenadores que había estudiado me comunicaba hasta los últimos detalles: capacidad de memoria, resistencia al calentamiento, posibilidades de empleo, intercambio de lenguajes, precio y plazos de pago, pesos, medidas, resoluciones de color, consumo, eficacia de los servicios técnicos de los importadores, apoyo de libros y revistas, usos más comunes en aquel momento... Durante horas fue alimentando aquel cerebro de elefante que El me había adjudicado y en el que confiaba tanto. E inmediatamente me preguntó cuál debía elegir.

—Negativo. Negativo —dije.

—¿Cuál eliges?— insistió. —El mejor.

—Negativo. Negativo.

—¿Ninguno es el mejor?

No hay respuesta.

Vi como se enfurecía y cerraba los puños.

Entre las informaciones que me había proporcionado, en el apartado de «costes», había deducido de cada modelo seleccionando la cantidad que, al adquirir cada uno de ellos, le descontaban si me entregaba a mí como parte del pago. Una cantidad siempre ridícula, pues aunque teóricamente viejo, mis servicios pueden ofrecerle aún buenos resultados...

—¿El mejor?— insistió.

—Frío— dije yo. —Frío.

—Elije uno, Colega.

Aquella palabra penetró como un torrente de sol en mi corazón de silicio. Y di respuesta a su última pregunta... Quizás encontrara yo a otro buen compañero que me librase de la helada muerte.



PERIFERICOS SPECTRUM A POSTERIORI

El **Spectrum** se rodea siempre de lo mejor.
Si, a priori, te decidiste por el más popular de los ordenadores, decídete, a posteriori, por los más importantes periféricos.

Interface 1
Permite la conexión de hasta ocho Microdrives, que pueden guardar hasta 680KB de datos y programas.
Tiene un canal RS232 para conectar impresoras, modems, ...
A través de su toma de red de área local puede comunicarse con otros 63 **Spectrum.**

Microdrive ZX
Para almacenar más de 85KB en cada cartucho magnético, con un tiempo de acceso de 3,5 segundos.

Interface 2
Para usar programas contenidos en cartuchos ROM. Contiene un controlador para dos Investicks.

Investick
Joystick de gran robustez, con mando ergonómico, ventosas para fijarlo y cuatro puntos de disparo, para disfrutar más con los juegos.

Interface programable con sonido
Permite usar los Investicks en **todos** los programas, con los efectos sonoros al volumen deseado.

Lápiz óptico
Para crear directamente en la pantalla todos los dibujos imaginables. Se puede dibujar, colorear, reducir/ampliar, mezclar con textos u otros dibujos, grabar y cargar pantallas. ...

SPECTRUM A PRIORI

DISTRIBUIDOR EXCLUSIVO **investronica**
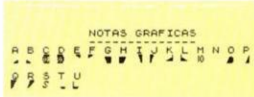
Tomás Bretón, 62 Tel. (91) 467 82 10 Telex 23399 IYCO E 28045 Madrid
Camp. 80 Tels. (93) 211 26 58 - 211 27 54 08022 Barcelona

etc

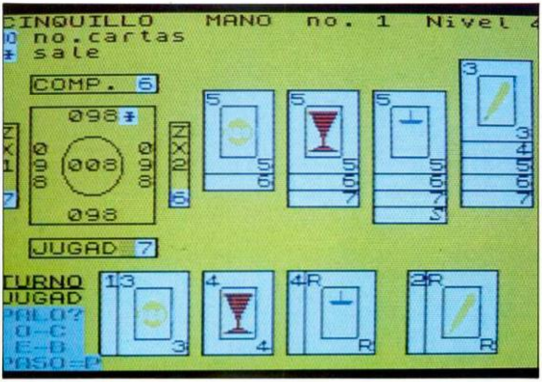
CINQUILLO

48 K

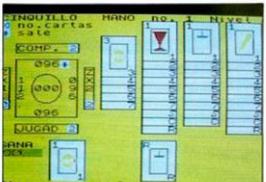
Basado en el tradicional y muy popular juego de naipes del mismo nombre, este programa te ofrece la opción de jugar tú solo contra el ordenador, siguiendo las mismas reglas del cinquillo.
Consiste en hacer una serie de escaleras partiendo de la carta cinco de cada palo.
Tras repartir diez cartas a cada jugador, ganará aquél que antes se quede sin cartas.
Hay cuatro niveles de dificultad que te permitirán, si eres habilidoso, hacerte con unas jugosas ganancias.
Anímate, es cuestión de jugar.




```
  10 GO TO 8900
 100 REM Inicio partida
 105 LET ayuda=0: LET mano=0: LET c$="      ": LET b$=c$+c$+c$
 110 DIM c(5): FOR i=1 TO 4: LET c(i)=100: NEXT i
 120 DEF FN v(x,y,z,u,w)=(x=0)*(ABS (w-5)+1+(w<5))+(x*(y*x+z)+(x=3 AND u=2)*3)*10
 125 DEF FN w(x,a,y,b,z,c,w)=1000-(ABS (w-x)+1)*((w-x)>=0)*10*a-ABS (w-x)*((w-x)<0)*100*a-(ABS (w-y)+1)*((w-y)>=0)*10*b-ABS (w-y)*((w-y)<0)*100*b-(ABS (w-z)+1)*((w-z)>=0)*10*c-ABS (w-z)*((w-z)<0)*100*c
 200 GO SUB 8500: GO SUB 7000
 205 DIM d(4,2): LET d(1,1)=5: LET d(1,2)=5: LET d(2,1)=15: LET d(2,2)=15: LET d(3,1)=25: LET d(3,2)=25: LET d(4,1)=35: LET d(4,2)=35
 210 LET jug=juma: LET carta=5: GO SUB 6920
 215 PRINT AT 16,0;"TURNO"; OVER 1;AT 16,0;"_____": GO TO 7300
 216 PAUSE 100
 220 LET jug=jug+1: IF jug=5 THEN LET jug=1
 225 PRINT BRIGHT 1;AT 17,0;j$(jug): BEEP .2,1: BEEP .2,10
 230 PRINT PAPER 6;AT 10,0;" ";AT 4,7;" ";AT 14,7;" ";AT 10,10;" ": PLOT 0,95: DRAW 0,-8
 235 LET a1=10*(jug=1 OR jug=3)+4*(jug=2)+14*(jug=4)
 240 LET a2=0*(jug=1)+7*(jug=2 OR jug=4)+10*(jug=3)
 245 PRINT PAPER 6; INK 2; FLASH 1;AT a1,a2;" "
 250 IF jug=4 AND demo=0 THEN GO TO 2000
 300 GO SUB 6200
 385 IF nuca=0 THEN GO TO 7405
 390 IF nuca=1 THEN GO TO 7305
 395 IF nivel=1 THEN GO TO 500
 400 LET nico=1: GO SUB 7800: IF nuca=1 THEN GO TO 7305
 405 IF nivel=2 THEN GO TO 500
 410 LET nico=2: GO SUB 7800: IF nuca=1 THEN GO TO 7305
 415 IF nivel=3 THEN GO TO 500
 420 LET nico=3: GO SUB 7800: IF nuca=1 THEN GO TO 7305
 430 GO TO 500
 500 REM elegir carta al azar
 505 DIM o(nuca): LET z=1
 510 FOR k=1 TO 4
 515 FOR i=1 TO 2
 520 IF e(k,i,1)=0 THEN GO TO 530
 525 IF NOT (i=2 AND d(k,1)=d(k,2)) THEN LET o(z)=d(k,i): LET z=z+1
 530 NEXT i
 535 NEXT k
 536 IF demo=1 THEN GO TO 545
 540 IF jug=4 AND ayuda=1 THEN GO TO 2040
 545 LET z=1+INT (RND*nuca): LET carta=o(z)
 550 GO TO 7305
2000 REM Juega Jugador
2005 LET ayuda=0
2010 PRINT FLASH 1; INK 2;AT 19,0;"AYUDA";AT 20,0;" 5/N?": BEEP .1,20
2015 IF INKEY$="" THEN GO TO 2015
2020 IF INKEY$="s" OR INKEY$="5" THEN GO TO 2030
2025 GO TO 2100
2030 PRINT AT 20,0;c$: LET ayuda=1: GO TO 300
2035 DIM o(1): LET o(1)=carta
2040 PRINT AT 20,0;c$
2045 FOR i=1 TO nuca
2050 LET carta=o(i): GO SUB 6920
2055 LET col=b(palo,2)-1
2060 FOR k=10*palo-9 TO 10*palo
2065 IF a(4,k)<>1 THEN GO TO 2075
2070 LET col=col+1: IF k=o(i) THEN PRINT OVER 1; PAPER 8; INK 8;: FLASH 1;AT 16,col;" "
2075 NEXT k
2080 NEXT i
2090 PRINT FLASH 1; INK 2;AT 20,0;"PULSA";AT 21,0;"TECLA": BEEP .1,10
2095 PAUSE 0
2100 PRINT AT 19,0;c$;AT 20,0;c$
2105 LET ayuda=0
2110 PRINT FLASH 1; INK 1; BRIGHT 1;AT 18,0;"PALO?";AT 19,0;" O-C ";AT 20,0;" E-B ";AT 21,0;"PASO=P": BEEP .1,10
2115 POKE 23658,8: REM MAYUSCULAS
2120 IF INKEY$="" THEN GO TO 2120
2121 PAUSE 0
2125 IF INKEY$="P" THEN GO TO 7405
2130 IF INKEY$="O" THEN LET palo=1: LET p$="oros": GO TO 2160
2135 IF INKEY$="C" THEN LET palo=2: LET p$="copas": GO TO 2160
2140 IF INKEY$="E" THEN LET palo=3: LET p$="espad": GO TO 2160
2145 IF INKEY$="B" THEN LET palo=4: LET p$="basto": GO TO 2160
2150 GO TO 2110
2155 PAUSE 15
2160 PRINT FLASH 1; BRIGHT 1;AT 18,0;"Carta";AT 19,0;"1,2..";AT 20,0;"S,C,R";AT 21,0;"  ?  ": PRINT AT 21,5;" ": IF b(1,1)<>0 THEN PLOT 47,7
2161 BEEP .1,10: POKE 23658,0: REM minusculas
2162 PAUSE 0
2165 IF INKEY$="" THEN GO TO 2165
2166 IF LEN INKEY$<>1 THEN GO TO 2160
2171 IF CODE INKEY$>48 AND CODE INKEY$<56 THEN LET k$=INKEY$: LET l$=" "+k$+" de": LET facial=VAL k$: GO TO 2200
2175 IF INKEY$="s" THEN LET l$="sota ": LET facial=8: LET k$="S": GO TO 2200
2180 IF INKEY$="c" THEN LET l$="cab. ": LET facial=9: LET k$="C": GO TO 2200
2185 IF INKEY$="r" THEN LET l$=" rey ": LET facial=10: LET k$="R": GO TO 2200
2190 GO TO 2160
2200 PRINT FLASH 1; BRIGHT 1;AT 18,0;l$;AT 19,0;p$;AT 20,0;" S/N?";AT 21,0;c$: BEEP .1,10
2204 IF INKEY$="" THEN GO TO 2204
2205 IF INKEY$="N" OR INKEY$="n" THEN GO TO 2110
2206 IF INKEY$="5" OR INKEY$="s" THEN GO TO 2210
2207 GO TO 2204
2210 PRINT OVER 1; INK 8; PAPER 8;AT 16,2;b$;b$
2215 LET carta=(palo-1)*10+facial
2220 IF a(4,carta)<>1 OR (d(palo,1)<>carta AND d(palo,2)<>carta) THEN GO TO 2300
2225 PRINT AT 18,0;c$;AT 19,0;c$;AT 20,0;c$;AT 21,0;c$
2230 GO TO 7305
2300 PRINT FLASH 1;AT 18,0;" NO ";AT 19,0;"VALE ": PAUSE 10
2305 GO TO 2090
2500 STOP
5999 REM Evaluacion intrinseca de las cartas
6000 DIM v(4,44)
6010 FOR k=1 TO 4: PRINT AT 20,12; FLASH 1;" DE ";j$(k);" ": BEEP .1,20: REM jugador
6011 FOR l=1 TO 4: REM palo
6015 FOR j=l*10-5 TO l*10: REM carta 5 a 10
6020 IF a(k,j)=0 THEN GO TO 6095
6050 LET kkk=1: LET fin=10
6055 GO SUB 6700
6070 IF facial=5 THEN LET v(k,40+l)=v(k,j)
6075 NEXT j
6080 FOR j=l*10-1 TO l*10-5 STEP -1
6085 IF a(k,j)=1 AND a(k,j+1)=1 THEN LET v(k,j)=v(k,j+1)
6090 IF j=l*10-5 THEN LET v(k,40+l)=v(k,j)
6095 NEXT j
6100 FOR j=l*10-5 TO l*10-9 STEP -1: REM carta 5 a 1
6105 IF a(k,j)=0 THEN GO TO 6145
6120 LET kkk=-1: LET fin=1
6130 GO SUB 6700
6145 NEXT j
6150 FOR j=l*10-8 TO l*10-5
6155 IF a(k,j)=1 AND a(k,j-1)=1 THEN LET v(k,j)=v(k,j-1)
6160 IF j=l*10-5 AND v(k,40+l)>v(k,j) THEN LET v(k,j)=v(k,40+l)
6165 NEXT j
6170 NEXT l
6175 NEXT k
6180 RETURN
6200 REM Valoracion cartas
6205 DIM e(4,2,3): LET nuca=0
6210 LET a1=jug+1-(jug=4)*4: LET a2=a1+1-(a1=4)*4: LET a3=a2+1-(a2=4)*4
6215 LET x=a(a3,41): LET y=a(a2,41): LET z=a(a1,41): LET w=a(jug,41)
6220 FOR k=1 TO 4
6225 FOR i=1 TO 2
6229 IF d(k,i)=0 THEN GO TO 6275
6230 IF a(jug,d(k,i))<>1 THEN GO TO 6275
6235 LET nuca=nuca+1: LET carta=d(k,i): IF i=2 AND d(k,1)=d(k,2) THEN LET nuca=nuca-1
6236 LET aux=INT (v(jug,d(k,i))/10): LET aux1=v(jug,d(k,i))-aux*10
6240 LET e(k,i,1)=(nivel<=2)*v(jug,d(k,i))+(nivel>2)*aux*10+(nivel>2)
6245 LET a=(a(a3,d(k,i))=-10): LET b=(a(a2,d(k,i))=-10): LET c=(a(a1,d(k,i))=-10)
6250 LET e(k,i,2)=FN w(x,a,y,b,z,c,w)+(nivel=3)*aux1
6255 LET a=d(k,i): IF ((a=1 OR a=11 OR a=21 OR a=31) AND i=1) OR ((a=10 OR a=20 OR a=30 OR a=40) AND i=2) THEN LET c=0: GO TO 6270
6256 LET a=a+(i=2)-(i=1)
6260 LET b=a(5,a): LET a=1: IF nivel=4 THEN LET a=(jug<>2)-(jug=2)
6265 LET c=(b=a2)*a-(b=a1)-(b=a3)
6270 LET e(k,i,3)=100-c+aux1*(nivel=4)
6275 NEXT i
6280 NEXT k
6285 RETURN
6700 REM valor intrinseco de una carta
6705 LET facial=j
6710 IF facial>10 THEN LET facial=facial-10: GO TO 6710
6715 LET ca=0: LET ga=0: LET ga1
```

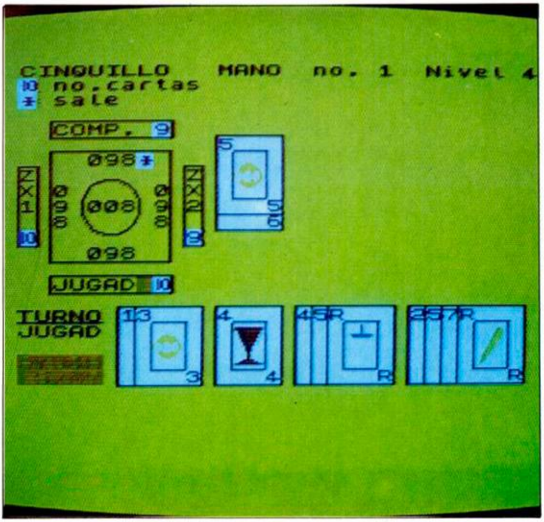


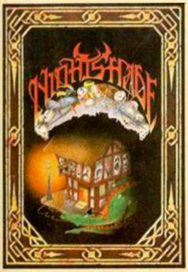

```
=0: LET gaind=0: LET ga2=0
6720 FOR i=facial+kkk TO fin STEP kkk
6721 LET co=i+10*(l-1)
6725 IF a(k,co)=1 THEN LET ca=ca+1: IF a(k,co-kkk)=0 THEN LET gaind=gaind+1
6730 IF a(k,co)=0 AND gaind=0 THEN LET ga1=ga1+1
6735 IF a(k,co)=0 AND gaind=1 THEN LET ga2=ga2+1
6740 IF a(k,co)=0 THEN LET ga=ga+1
6745 NEXT i
6750 LET v(k,j)=FN v(ca,ga1,ga2,gaind,facial)
6755 RETURN
6800 LET li=16
6805 FOR h=1 TO 4
6810 GO SUB 6825
6815 NEXT h
6816 RETURN
6820 REM dibujar el palo h del jugador
6825 LET col=b(h,2)-1
6830 FOR k=10*h-9 TO 10*h
6835 IF a(4,k)<>1 THEN GO TO 6845
6840 LET col=col+1: LET carta=k: GO SUB 6920: GO SUB 8000
6845 NEXT k
6850 RETURN
6900 PRINT AT 17,15;"ESTOY": RETURN
6905 PRINT AT 19,12;"LAS  CARTAS": RETURN
6910 PRINT AT 17,12;b$;AT 18,12;b$;AT 19,12;b$: RETURN
6920 LET palo=1*(carta<11)+2*(carta>10 AND carta<21)+3*(carta<31 AND carta>20)+4*(carta>30)
6921 LET facial=carta
6925 IF facial>10 THEN LET facial=facial-10: GO TO 6925
6930 IF facial<8 THEN LET k$=STR$ facial: RETURN
6931 IF facial=8 THEN LET k$="S": RETURN
6932 IF facial=9 THEN LET k$="C": RETURN
6933 IF facial=10 THEN LET k$="R": RETURN
```




```
6970 LET ccc=c(1): GO SUB 6995: PRINT BRIGHT 1; PAPER 6;AT 8,2;a1;AT 9,2;a2;AT 10,2;a3: RETURN
6975 LET ccc=c(2): GO SUB 6995: PRINT BRIGHT 1; PAPER 6;AT 6,4;a1;AT 6,5;a2;AT 6,6;a3: RETURN
6980 LET ccc=c(3): GO SUB 6995: PRINT BRIGHT 1; PAPER 6;AT 8,8;a1;AT 9,8;a2;AT 10,8;a3: RETURN
6985 LET ccc=c(4): GO SUB 6995: PRINT BRIGHT 1; PAPER 6;AT 12,4;a1;AT 12,5;a2;AT 12,6;a3: RETURN
6990 LET ccc=c(5): GO SUB 6995: PRINT BRIGHT 1; PAPER 6;AT 9,4;a1;AT 9,5;a2;AT 9,6;a3: RETURN
6995 LET a3=ccc-10*INT (ccc/10): LET a1=INT (ccc/100): LET a2=INT ((ccc-100*a1)/10): RETURN
7000 REM Barajando
7001 LET mano=mano+1: PRINT AT 0,12;"MANO  no. ";mano
7005 GO SUB 6900: PRINT AT 18,13; FLASH 1;"BARAJANDO": GO SUB 6905
7006 BEEP .1,8: BEEP .3,15
7010 DIM a(5,41): DIM b(4,2)
7015 FOR i=1 TO 40: LET a(4,i)=1: NEXT i
7020 LET c(5)=0
7025 FOR i=1 TO 4
7030 LET c(i)=c(i)-2: LET c(5)=c(5)+2: GO SUB (6965+5*i): BEEP .2,5: GO SUB 6990: BEEP .2,8
7035 NEXT i
7045 PAUSE 40: GO SUB 6910
7050 PRINT AT 1,0; PAPER 5; BRIGHT 1;"10";AT 1,2; PAPER 4; BRIGHT 0;"no.cartas"
7100 REM Repartiendo
7105 RANDOMIZE
7110 GO SUB 6900: PRINT AT 18,12; FLASH 1;"REPARTIENDO": GO SUB 6905
7111 BEEP .1,8: BEEP .3,15
7115 FOR j=1 TO 10
7120 FOR i=1 TO 3
7125 LET carta=INT (RND*40)+1
7130 IF a(4,carta)<>1 THEN GO TO 7125
7135 LET a(4,carta)=0: LET a(i,carta)=1
7140 NEXT i
7145 NEXT j
7150 FOR i=1 TO 4: LET b(i,1)=0
7151 FOR j=10*i-9 TO 10*i
7152 IF a(4,j)=1 THEN LET b(i,1)=b(i,1)+1
7155 NEXT j
7160 NEXT i
7175 LET a4=28-b(4,1): LET a3=a4-b(3,1)+(b(3,1)<>0)-5*(b(4,1)<>0): LET a2=a3-b(2,1)+(b(2,1)<>0)-5*(b(3,1)<>0): LET a1=a2-b(1,1)+(b(1,1)<>0)-5*(b(2,1)<>0)
7180 LET juma=0: LET b(4,2)=a4: LET b(3,2)=a3: LET b(2,2)=a2: LET b(1,2)=a1
7185 PRINT PAPER 5; BRIGHT 1;AT 11,0;"10";AT 11,10;"10";AT 4,8;"10";AT 14,8;"10": PLOT 0,87: DRAW 0,-8
7190 FOR i=1 TO 4
7195 LET a(i,41)=10: IF a(i,5)=1 THEN LET juma=i
7200 NEXT i
7205 PRINT AT 18,12; FLASH 1;" ORDENANDO ": BEEP .1,8: BEEP .3,15: GO SUB 6000: GO SUB 6910: PRINT AT 20,12;b$: LET li=16
7210 FOR h=1 TO 4: GO SUB 6825: NEXT h
7215 PRINT PAPER 5;AT 2,0;"*": PRINT AT 2,2;"sale"
7220 IF juma=1 THEN PRINT PAPER 5;AT 7,2;"*"
7225 IF juma=2 THEN PRINT PAPER 5;AT 6,7;"*"
7230 IF juma=3 THEN PRINT PAPER 5;AT 11,8;"*"
7235 IF juma=4 THEN PRINT PAPER 5;AT 12,3;"*"
7240 RETURN
7300 REM Poner carta
7305 GO SUB 6920
7306 IF demo=1 THEN GO TO 7310
7307 IF jug=4 AND ayuda=1 THEN GO TO 2035
7310 LET a(jug,carta)=-1: LET a(5,carta)=jug: LET a(jug,41)=a(jug,41)-1
7315 LET col=7+palo*5: LET li=facial
7320 IF facial<=5 THEN LET d(palo,1)=carta-1: IF k$="1" THEN LET d(palo,1)=0
7325 IF facial>=5 THEN LET d(palo,2)=CARTA+1: IF k$="R" THEN LET d(palo,2)=0
7330 IF facial<4 THEN GO SUB 8000: GO TO 7335
7331 FOR k=10 TO 1 STEP -1
7332 LET carta=(palo-1)*10+k: IF a(5,carta)<>1 AND a(5,carta)<>2 AND a(5,carta)<>3 AND a(5,carta)<>4 THEN GO TO 7334
7333 GO SUB 6920: LET li=facial: GO SUB 8000
7334 NEXT k
7335 LET ca1=(jug=1 OR jug=3)*11+(jug=2)*4+(jug=4)*14: LET ca2=(jug=1)*0+(jug=2 OR jug=4)*8+(jug=3)*10
7340 PRINT BRIGHT 1; PAPER 5;AT ca1,ca2;a(jug,41): IF jug=1 THEN PLOT 0,87: DRAW 0,-8
7345 IF jug<>4 THEN GO TO 7365
7350 PLOT INVERSE 1;8*b(palo,2)-1,48: DRAW INVERSE 1;8*b(palo,1)+25,0: DRAW INVERSE 1;0,-41: DRAW INVERSE 1;-8*b(palo,1)-25,0: DRAW INVERSE 1;0,41
7355 FOR j=1 TO b(palo,1): FOR i=1 TO 5: PRINT AT 15+i,b(palo,2)-1+j;" ": NEXT i: NEXT j
7360 LET h=palo: LET li=16: LET b(palo,1)=b(palo,1)-1: IF b(palo,1)<>0 THEN GO SUB 6825
7365 IF a(jug,41)=0 THEN LET gan=jug: GO TO 7500
7370 GO TO 220
7400 REM Paso
7401 IF demo=1 THEN GO TO 7409
7405 IF jug=4 AND ayuda=1 THEN LET ayuda=0: PRINT AT 20,0;c$
7409 PRINT AT 18,0;c$;AT 19,0;c$;AT 20,0;c$;AT 21,0;c$;" ": IF b(1,1)<>0 THEN PLOT 47,7
7410 PRINT FLASH 1;AT 19,0;"PASO": BEEP .1,.1: BEEP .6,-10
7415 LET c(jug)=c(jug)-1: LET c(5)=c(5)+1
7420 GO SUB 6965+5*jug: BEEP .2,5: GO SUB 6990: BEEP .2,8
7425 PRINT AT 19,0;c$
7430 FOR i=1 TO 2: FOR k=1 TO 4
7435 IF d(k,i)<>0 THEN LET a(jug,d(k,i))=-10
7440 NEXT k
7445 NEXT i
7450 GO TO 220
7500 REM Fin de la mano
7505 PRINT AT 16,0;"GANA "' FLASH 1;j$(gan)' FLASH 0;c$'c$'c$
7510 GO SUB 9010
7515 LET k=0: LET c(gan)=c(gan)+c(5): LET c(5)=0: GO SUB 6965+5*gan: GO SUB 6990
7520 FOR i=1 TO 4
7525 IF c(i)<12 THEN LET k=1
7530 NEXT i
7540 IF k<>0 THEN GO TO 7600
7545 IF demo=1 THEN PAUSE 100: GO TO 200
7550 PRINT AT 21,0;"Otra mano s/n ?": IF INKEY$="" THEN GO TO 7550
7555 IF INKEY$="n" OR INKEY$="N" THEN GO TO 7600
7560 IF INKEY$="s" OR INKEY$="S" THEN GO TO 200
7565 GO TO 7550
7600 REM Fin de la partida
7605 LET k=0: LET j=1E6: GO SUB 8560
7610 PRINT AT 0,10;"RESULTADOS"; OVER 1;AT 0,10;"__________"
7615 FOR i=1 TO 4
7620 PRINT AT i+1,6;j$(i);" = ";c(i)
7624 IF c(i)=k THEN LET gan=0: GO TO 7630
7625 IF c(i)>k THEN LET k=c(i): LET gan=i
7630 IF c(i)<j THEN LET j=c(i): LET per=i
7635 NEXT i
7636 IF gan=0 THEN GO TO 7650
7640 PRINT AT 10,10;"GANADOR :";AT 11,11; FLASH 1;j$(gan)
7645 IF gan=4 THEN GO SUB 9020: GO TO 7660
7650 IF per=4 THEN GO SUB 9000: PRINT FLASH 1;AT 14,10;"PERDISTE": GO TO 7660
7655 GO SUB 9010
7660 PAUSE 0
7670 STOP
7700 IF demo=1 THEN LET nivel=2:
```

```
 GO TO 7735
7701 PRINT AT 17,10;"ESCOGER NIVEL";AT 18,11;"DE DIFICULTAD";AT 19,13;"(1 A 4)"
7710 IF INKEY$="" THEN BEEP .5,10: PAUSE 30: GO TO 7710
7715 IF LEN INKEY$<>1 THEN GO TO 7705
7720 IF CODE INKEY$<49 OR CODE INKEY$>52 THEN GO TO 7705
7725 LET nivel=CODE INKEY$-48
7730 PRINT AT 17,10;b$;AT 18,10;b$;AT 19,13;b$
7735 PRINT AT 0,25;"Nivel ";nivel
7740 RETURN
7800 REM Hallar maxima valoracion de una carta
7801 LET val=-10
7805 FOR k=1 TO 4
7810 FOR i=1 TO 2
7820 IF e(k,i,nico)>val THEN LET val=e(k,i,nico): LET carta=d(k,i)
7825 NEXT i
7830 NEXT k
7835 FOR k=1 TO 4
7840 FOR i=1 TO 2
7845 IF d(k,i)=0 THEN GO TO 7860
7850 IF e(k,i,nico)<val AND e(k,i,nico)<>0 THEN LET nuca=nuca-1: IF i=2 AND d(k,1)=d(k,2) THEN LET nuca=nuca+1
7855 IF e(k,i,nico)<val THEN LET e(k,i,1)=0: LET e(k,i,2)=0: LET e(k,i,3)=0
7860 NEXT i
7865 NEXT k
7870 RETURN
8000 REM Dibujo carta
8001 INK 0: LET a1=8*col: LET a2=159-8*li
8002 BEEP .05,30
8005 FOR i=0 TO 4: PRINT PAPER 7;AT li+i,col;"    ": NEXT i
8010 PLOT a1-1,a2+17: DRAW 33,0: DRAW 0,-41: DRAW -33,0: DRAW 0,41
8015 PLOT a1+7,a2+9: DRAW 17,0: DRAW 0,-25: DRAW -17,0: DRAW 0,25
8020 IF palo=1 THEN PRINT PAPER 7; INK 6;AT li+1,col+1;"▁";AT li+2,col+1;"▆";AT li+3,col+1;"▔"
8025 IF palo=2 THEN PRINT PAPER 7; INK 2;AT li+1,col+1;"▀";AT li+2,col+1;"▼";AT li+3,col+1;"▲"
8030 IF palo=4 THEN PRINT PAPER 7; INK 4;AT li+1,col+2;"▗";AT li+2,col+1;"▞";AT li+3,col+1;"▞": PLOT INK 4;a1+16,a2-8
8035 IF palo=3 THEN PRINT PAPER 7; INK 1;AT li+1,col+1;"┴": INK 5: PLOT a1+15,a2: DRAW 0,-13: PLOT a1+16,a2: DRAW 0,-14: PLOT a1+17,a2: DRAW 0,-12: INK 0
8045 PRINT PAPER 7;AT li,col;k$;AT li+4,col+3;k$
8050 RETURN
8400 REM Graficos
8405 RESTORE : FOR i=1 TO 20
8410 READ k$
8415 FOR j=0 TO 7: READ k: POKE USR k$+j,k: NEXT j
8420 NEXT i
8421 DIM j$(4,5): LET j$(1)="ZX1  ": LET j$(2)="COMP.": LET j$(3)="ZX2  ": LET j$(4)="JUGAD"
8425 RETURN
8430 DATA "a",0,0,0,0,0,3,15,31,"b",0,0,0,0,0,192,240,248
8431 DATA "c",63,63,108,85,84,85,45,63,"d",252,252,238,86,214,86,108,252
8432 DATA "e",31,15,3,0,0,0,0,0,"f",248,240,192,0,0,0,0,0
8435 DATA "g",0,0,127,127,63,63,31,31,"h",0,0,254,254,252,252,248,248,"i",15,15,7,7,3,3,3,1
8436 DATA "j",240,240,224,224,192,192,192,128,"k",1,1,1,7,31,127,0,0,"l",128,128,128,224,248,254,0,0
8440 DATA "m",0,38,41,41,41,41,36,0
8441 DATA "o",0,0,28,62,62,126,126,252,"p",0,1,1,3,3,7,7,15,"q",252,248,240,240,224,224,192,128,"r",15,31,30,28,56,48,96,0
8445 DATA "t",0,0,0,0,0,0,31,15,"u",0,128,128,128,128,128,252,248
8450 DATA "s",0,30,34,16,8,68,120,0
8500 REM Preparar la mesa
8505 PAPER 4: BORDER 4: INK 0: GO SUB 8560: PRINT AT 0,0;"CINQUILLO"
8510 FOR i=6 TO 12: PRINT PAPER 6;AT i,2;"       ": NEXT i
8515 PLOT 15,128: DRAW 57,0: DRAW 0,-57: DRAW -57,0: DRAW 0,57
8520 PRINT PAPER 6;AT 4,2;j$(2);" "; BRIGHT 1;"  ": PLOT 15,144: DRAW 58,0: DRAW 0,-9: DRAW -58,0: DRAW 0,9
8525 PRINT PAPER 6;AT 14,2;j$(4);" "; BRIGHT 1;"  ": PLOT 15,64: DRAW 58,0: DRAW 0,-9: DRAW -58,0: DRAW 0,9
8530 PRINT PAPER 6;AT 7,0;"Z";AT 8,0;"X";AT 9,0;"1";AT 10,0;" ";AT 11,0; BRIGHT 1;" ": PLOT 0,120: DRAW 9,0: DRAW 0,-41: DRAW -9,0: DRAW 0,41
8535 PRINT PAPER 6;AT 7,10;"Z";AT 8,10;"X";AT 9,10;"2";AT 10,10;" ";AT 11,10; BRIGHT 1;" ": PLOT 79,120: DRAW 9,0: DRAW 0,-41: DRAW -9,0: DRAW 0,41
8540 CIRCLE 43,99,14
8545 LET c(5)=0: FOR i=1 TO 5: GO SUB (6965+i*5): NEXT i
8550 GO SUB 7700
8555 RETURN
8560 CLS : RETURN
8900 REM Presentacion
8905 PAPER 6: INK 0: GO SUB 8560
8910 GO SUB 8400
8920 LET demo=0
8925 GO TO 9200
8930 PRINT BRIGHT 1; FLASH 1;AT 18,10;"PULSAR TECLA"; FLASH 0;AT 19,8;"I= INSTRUCCIONES";AT 20,8;"D= DEMOSTRACION ";AT 21,8;"J= JUGAR         "
8935 IF INKEY$="" THEN GO TO 8935
8940 IF INKEY$="j" OR INKEY$="J" THEN GO TO 100
8945 IF INKEY$="i" OR INKEY$="I" THEN GO TO 9100
8950 IF INKEY$="d" OR INKEY$="D" THEN LET demo=1: GO TO 100
8960 GO TO 8930
9000 REM RIP
9001 RESTORE 9050: FOR l=1 TO 24: READ a,b: BEEP a,b: NEXT l
9002 RETURN
9010 REM Acierto
9011 RESTORE 9060: FOR l=1 TO 11: READ a,b: BEEP a,b: NEXT l
9012 RETURN
9020 REM ganador
9021 FOR o=6 TO 10: BEEP 1,o: NEXT o
9022 FOR o=1 TO 4: GO SUB 9024: NEXT o
9023 RETURN
9024 RESTORE 9065: FOR l=1 TO 9: READ a,b: BEEP a,b: NEXT l: RETURN
9050 DATA 1,-1.7,1,-1.7,.2,-1.7,1,-1.7,.5,1.4,.5,.35,.5,.35,.5,-1.7,1,-1.7,.2,-1.7,1,-1.7
9055 DATA .2,-2,.2,1,.5,1,.2,-1,.2,3,.5,3,.2,5,.2,5,.2,3,.2,5,.2,6,.2,8,.6,10
9060 DATA .2,1,.09,-60,.1,-4,:1,-4,.1,-4,.4,-3,.58,-4,.1,-60,.3,0,.09,-60,.2,1
9065 DATA .1,14,.1,12,.4,11,.1,11,.1,11,.1,9,.1,11,.4,12,.3,11
9095 PRINT AT 21,0; FLASH 1;"PULSAR UNA TECLA PARA CONTINUAR": PAUSE 0: RETURN
9100 REM instrucciones
9105 PAPER 6: INK 0: GO SUB 8560
9110 PRINT AT 0,9;"INSTRUCCIONES";AT 0,9; OVER 1;"_____________": PRINT
9115 PRINT AT 2,2;"El cinquillo consiste en hacerescaleras de cada uno de los cuatro palos."
9120 PRINT AT 5,2;"Se reparten diez cartas a cadajugador y este pone en  el platodos puntos para participar.(Cadajugador tiene inicialmente  cienpuntos)."
9125 PRINT AT 10,2;"Sale el cinco de oros y el turno corre siempre  al  jugador dela derecha."
9130 PRINT AT 13,2;"Cada jugador pone una carta para continuar,por arriba o  por abajo, una escalera empezada o uncinco para empezar una  nueva escalera."
9135 PRINT AT 18,2;"Si no puede poner una carta,pasa y pone un punto en el plato."
9140 GO SUB 9095
9145 GO SUB 8560
9150 PRINT AT 2,2;"El primer jugador que se quedasin cartas, gana y se  lleva  elplato."
9155 PRINT AT 5,2;"Hay cuatro niveles de  dificultad creciente."
9160 PRINT AT 8,9;"BUENA SUERTE!!!"
9165 GO TO 8930
9200 REM portada
9205 BORDER 6: PAPER 6: CLS
9210 LET li=1: LET col=1: LET palo=1: LET k$="5": GO SUB 8000
9220 LET li=1: LET col=27: LET palo=2: LET k$="5": GO SUB 8000
9230 LET li=16: LET col=1: LET palo=3: LET k$="5": GO SUB 8000
9240 LET li=16: LET col=27: LET palo=4: LET k$="5": GO SUB 8000
9250 PRINT AT 1,8;"▀▄";AT 2,7;"▌";AT 3,7;"▌";AT 4,8;"▀";AT 2,9;"▪";AT 3,9;"▌";AT 4,9;"▪"
9255 PRINT AT 3,10;"▌";AT 4,10;"▪";AT 2,11;"▄";AT 3,12;"▌";AT 4,12;"▪";AT 3,13;"▌";AT 2,13;"▄";AT 4,13;"▀";AT 3,14;"▌";AT 4,14;"▜"
9260 PRINT AT 2,15;"▖";AT 3,15;"▌";AT 4,16;"▀";AT 2,17;"▖";AT 3,17;"▌";AT 2,18;"▪";AT 3,18;"▌";AT 4,18;"▪"
9265 PRINT AT 2,19;"▌▌";AT 3,19;"▌▌";AT 4,19;"▪▪";AT 2,21;"▄";AT 3,21;"▌▌";AT 4,21;"▀"
9270 PRINT AT 7,10;"© JXER 1985"
9300 GO TO 8930
```

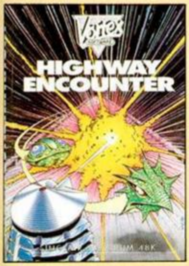






Lo último de ULTIMATE. Esta vez en una misteriosa ciudad encantada con los gráficos y animación que sólo ULTIMATE sabe dar.

NIGHTSHADE
Zx Spectrum
2.100 pts.



NOVEDADES
NOVEDADES
NOVEDADES
NOVEDADES
NOVEDADES
NOVEDADES
NOVEDADES
NOVEDADES
NOVEDADES
NOVEDADES

abc soft

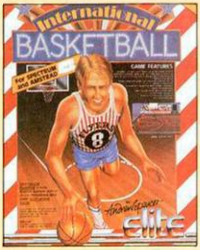
Santa Cruz de Marcenado, 31
28015 MADRID
Teléfs. (91) 248 82 13
(91) 242 50 59

NOVEDADES
NOVEDADES
NOVEDADES
NOVEDADES
NOVEDADES
NOVEDADES
NOVEDADES



Conduce a tus cinco vortons hasta la «Zona Cero» por la única vía libre que queda en la Tierra. Todo ello en un excelente escenario tridimensional que proporciona un realismo y animación sin límites.

HIGHWAY ENCOUNTER
Zx Spectrum y Amstrad 1.900 pts.



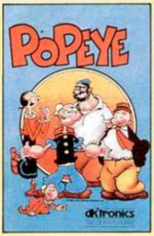
Fabuloso partido de baloncesto en el que podrás realizar y ensayar los más difíciles tiros que jamás hayas pensado.

INTERNATIONAL
BASKETBALL
Zx Spectrum y Amstrad
2.100 pts.



Un duro y sangriento combate de boxeo en el que deberás vencer a ocho expertos de este deporte... ¿Te animas a probar?

FRANK BRUNO'S
BOXING
Zx Spectrum
1.900 pts.





Un fabuloso juego en el que tú eres Popeye y habrás de demostrar tu amor a Oliva recogiendo 25 corazones. Todo ello acompañado de una fabulosa ambientación con todos los personajes del comic.

POPEYE * Zx Spectrum, C64, Amstrad y MSX

1.950 pts.

Si deseas recibir más información y propaganda de nuestros programas y de nuestras interesantes ofertas, por favor, rellena y envíanos este cupón.

Nombre .... Edad ....

Dirección ....

Localidad-Provincia ....

Ordenador .... Programas favoritos ....

La tortuga: Gráficos en Pascal



Alejandro Culvez y Marcos

Vamos a ver como a partir de la famosa tortuga del lenguaje LOGO, el PASCAL consigue enriquecerse con la posibilidad de crear gráficos.



¿Quién no conoce las instrucciones DRAW, PLOT, CIRCLE, etc.? ¿Quién no ha hecho alguna vez un pequeño dibujo utilizando dichas instrucciones? El BASIC nos da la posibilidad de crear gráficos sencillos utilizando las instrucciones antes mencionadas, pero no todos los lenguajes de programación tienen instrucciones dedicadas a la realización de gráficos. Entre estos lenguajes se encuentra el PASCAL.

El lenguaje PASCAL fue realizado en 1971 por Niklaus Wirth. Es un lenguaje que aporta a la programación nuevos conceptos como pueden ser la utilización de estructuras dinámicas de información (apuntadores), posibilidad de crear tipos de datos definidos por el usuario, posibilidad de crear registros variantes, etc, pero que adolece de instrucciones para la ralización de gráficos. Para subsanar esta falta, los usuarios del Spectrum que trabajan con el compilador PASCAL que existe en el mercado (HISOFT PASCAL) se encuentran con un paquete de procedimientos (rutinas), que basados en la tortuga del lenguaje LOGO, permiten la posibilidad de realizar gráficos de la misma manera que lo podrían hacer en BASIC, y que se conoce con el nombre de TURTLE GRAPHICS.

La tortuga del LOGO

El lenguaje LOGO, creado a finales de los años sesenta por el MIT (Instituto de Tecnología de Massachusetts) encabezados por Seymour Papert, es un lenguaje ideado pensando en el aprendizaje de los niños. *Una característica de este lenguaje* es la llamada «TORTUGA».

La tortuga es un robot que se arrastra por el suelo el cual lleva adosada una pluma que puede bajarse o subirse, según se quiera dibujar o no, respectivamente. El manejo de dicho robot es fácil, ya que sólo basta con mover el ingenio mecánico de un lado a otro para conseguir un bonito dibujo.

Como los pequeños ordenadores domésticos no pueden llevar adosada esta tortuga robótica, se simula colocando en la pantalla una «tortuga» (puede ser un rombo, un cuadrado, etc.), la cual iremos moviendo mediante un programa en lenguaje LOGO.

A continuación veremos cuáles son las instrucciones que manejan a la «tortuga» y el efecto que producen en la pantalla.

Comenzaremos por las instrucciones PENUP y PENDOWN. La instrucción PENUP, como su propio nombre indica, lo que hace es levantar el lápiz, es decir, deja de dibujar, mientras que la instrucción PENDOWN lo que hace es bajar el lápiz, es decir, que la tortuga pueda dibujar de nuevo.

La instrucción FORWARD n, lo que hace es avanzar la tortuga n unidades hacia delante, es decir, si, por ejemplo, tenemos que la toturga está posicionada en la posición (20,20) de la pantalla y queremos moverla a la posición (40,20), haremos:

FORWARD 20

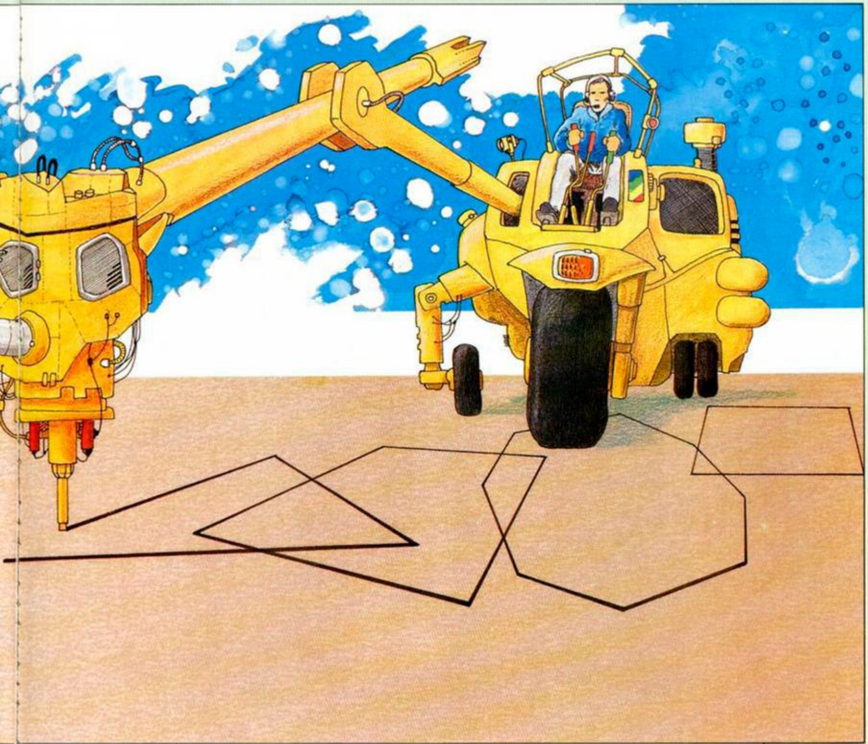
Pero esto tendrá razón de ser cuando la tortuga esté apuntando al este, es decir, cuando la tortuga esté orientada horizontalmente hacia la derecha. Supongamos que la tortuga está apuntando hacia el norte (hacia arriba) y queremos hacer el mismo movimiento que antes. Lo primero que habría que hacer es girar la tortuga 90 grados hacia la derecha o 270 grados hacia la izquierda. Esto se consigue con las instrucciones:

RIGHT n: volver hacia la derecha n grados

LEFT n: volver hacia la izquierda n grados

Así, el ejemplo anterior quedaría resuelto con las siguientes dos instrucciones:

RIGHT 90
FORWARD 20



Otra instrucción que se utiliza para el manejo de la tortuga es:

BACK n

Esta instrucción lo que hace es retroceder la tortuga n unidades hacia atrás:

Estas han sido las principales instrucciones que hay en LOGO para la realización de gráficos y que han sido implementadas en lenguaje PASCAL mediante procedimientos. Pero ¿qué es un procedimiento en PASCAL?

Los procedimientos en PASCAL

Cuando nosotros hacemos en BASIC una sentencia GOSUB n, lo que hacemos es ir a la instrucción que tiene el número n y una vez ejecutadas todas las instrucciones desde la n hasta la return, volver a la siguiente línea a la gosub. A ese grupo de instrucciones se les llama subrutinas.

El procedimiento en Pascal es semejante a la subrutina en Basic y tiene la siguiente sintaxis:

PROCEDURE nombre (parámetros);

DECLARACION;

(declaración de variables)

(declaración de constantes)

cuerpo;

Los parámetros se dividen en dos clases:

I. PARAMETROS VALOR

II. PARAMETROS VARIABLES

Ambos son nombres de variable, las cuales reciben un valor al producirse la llamada al procedimiento (parámetros valor), o dan un valor al programa al terminarse de ejecutar el procedimiento (parámetros variables).

Veamos un ejemplo:

Supongamos que queremos un procedimiento que nos calcule el producto de dos matrices de tal forma que el resultado lo depositemos en otra matriz. La cabecera del procedimiento puede ser la siguiente:

PROCEDURE PROMATRICES (A, B: MATRIZ; VAR C: MATRIZ);

¿Qué es lo que decimos con esto? Lo que queremos decir es que cuando se produzca la llamada al procedi-

miento nosotros le pasaremos dos parámetros de tipo valor que son la matriz A y B para que realize el producto y nos lo deposite en otra matriz C que es donde se encuentra el resultado de la ejecución del procedimiento. La llamada al procedimiento será la siguiente: PROMATRICES (MATRIZ, MATRIZII, PRODUCTO);

Ha de observarse que los nombres no tienen porque ser iguales, aunque si queremos pueden serlo.

Esta pequeña introducción al Pascal, más concretamente a los procedimientos, sirve únicamente para la correcta comprensión del paquete de procedimientos TURTLE GRAPHICS.

Turtle Graphics

Turtle Graphics es un paquete de procedimientos escrito en lenguaje Pascal, disponible para aquellos de vosotros que estéis en disposición del programa (HISOFT PASCAL).

Este conjunto de procedimientos que constituye el TURTLE GRAPHICS se carga desde el editor con el comando G,, TURTLE.

Debe tenerse en cuenta que cuando se carga el TURTLE no debe haber ningún programa residente. Una vez cargado el paquete vamos a ver los pasos necesarios para su correcta utilización:

I. El paquete ocupa desde la línea 10 hasta la línea 1350.

II. Debe crearse obligatoriamente antes de la línea 10 una instrucción PROGRAM y una instrucción VAR.

III. Si las necesitamos crearemos nuestras variables, constantes, etiquetas, declaraciones de tipos, etc. Si creamos algún procedimiento o función este debe situarse detrás del paquete TURTLE.

IV. A partir de la línea 1360 crearemos nuestro programa, o a partir de la línea correspondiente en caso de haber introducido nosotros algún procedimiento o función.

Un ejemplo aclara todos estos pasos:

```
1 PROGRAM EJEMPLO;
2 VAR
3 CONTADORRECTAS, CONTADORCURVAS = INTEGER;

[      ] →Paquete Turtle
1360 BEGIN
```

A continuación pasamos a describir todos y cada uno de los procedimientos que constituyen el TURTLE GRAPHICS.

Antes debemos hacer mención a 4 variables globales que aparecen y que son de vital importancia. Estas variables son las siguientes:

I. HEADING: sirve para guardar el valor angular de la dirección donde apunta la tortuga. Esta variable es de tipo real y es inicializada con el valor 0 cuando hacemos llamada al procedimiento Trutle.

El valor 0 significa que la tortuga apunta al este según el siguiente gráfico:

II. XCOR, YCOR: son las coordenadas de la tortuga en la pantalla. Como ya sabemos la pantalla tiene de dimensiones 256 x 176 pixels. Son variables de tipo real.

III. PENSTATUS: esta variable de tipo entero sirve para indicar el estado de la pluma (0-bajada, 1-subida).

Pasemos ahora tras esta breve descripción de las variables a describir los procedimientos de que consta el paquete TURTLE.

TURTLE: sirve para inicializar las variables del paquete de la siguiente manera;

Es conveniente utilizar este procedimiento y antes que cualquier otro debe utilizarse éste.

INK (C: INTEGER): procedimiento que sirve para cambiar el color de la tinta que vendrá dado por la variable C.

COPY: procedimiento que sirve para hacer una copia de la pantalla a impresora. En este procedimiento hay un error en la línea 570 donde se tiene USER (# EAC) y donde debe ponerse

INLINE (#FD, #21, #3A, #5C, # CD, #AC, #0E, # F3)

PENDOWN (C: INTEGER): procedimiento que sirve para bajar la pluma y así poder dibujar, además de poner tinta con la variable C.

PENUP: sirve para levantar la pluma y así dejar de dibujar.

SETHD (A: REAL): procedimiento que sirve para cambiar la orientación de la tortuga, recordando que si queremos que la tortuga apunte al norte haremos SETHD (90), si queremos que apunte al este SETHD (0), si queremos que apunte al oeste SETHD (180) y al sur SETHD (270).

SETXY (X, Y: REAL): posición de la tortuga a partir de los valores X e Y.

FWD (L: REAL): procedimiento que simula la instrucción FORWARD del LOGO, es decir, sirve para avanzar la tortuga L unidades en la dirección HEADING hacia adelante.

BACK (L:REAL): procedimiento que simula la instrucción BACK del LOGO, es decir, retrocede L unidades en la dirección HEADING hacia detrás.

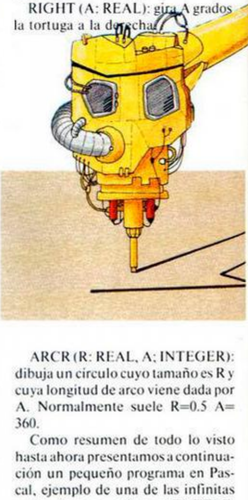
TURN (L:REAL): cambia la dirección de la tortuga A grados a partir de la posición en que se encuentre. Lo hace en sentido contrario a las agujas del reloj. Veamos un ejemplo: supongamos que la tortuga está colocada al norte 90 grados y que queremos que apunte:

a 120 grados noroeste: TURN (30)
a 65 grados noreste: TURN (−25)

VECTOR (A, L: REAL): procedimiento que desplaza la tortuga L unidades en la dirección de A.

LEFT (A:REAL): gira A grados la tortuga donde esté situada a la izquierda.

RIGHT (A: REAL): gira A grados la tortuga a la derecha.



ARCR (R: REAL, A; INTEGER): dibuja un círculo cuyo tamaño es R y cuya longitud de arco viene dada por A. Normalmente suele R=0.5 A=360.

Como resumen de todo lo visto hasta ahora presentamos a continuación un pequeño programa en Pascal, ejemplo de una de las infinitas aplicaciones que tiene el TURTLE, como veréis el TURTLE está a vuestra disposición.

Este programa construye una circunferencia a partir de la unión de 12 punto para ello disponemos de un cuadrado de lado=30. El valor del lado puede ser otro para ello basta con pasarle al procedimiento otro valor como luego indicaremos (Prog. 1).

Para aquellos de vosotros que no dispongáis del programa Pascal y por consiguiente del TURTLE, hemos incorporado un programa en lenguaje Basic que realiza exactamente las mismas funciones que el procedimiento TURTLE.

El listado de este programa aparece más adelante, pero es necesario explicar antes cómo realizamos el paso de parámetros a las subrutinas que es distinto a como lo realizamos en Pascal.

La subrutina FORWARD debe recibir como parámetro la variable L luego antes de hacer una llamada deberemos asignar a esta variable el valor correspondiente a dicha variable. Ej. LET L=20: GO SUB FORWARD.

Igualmente y en cada caso concreto debe procederse de esta manera para el paso de parámetros. A continuación describo los parámetros necesarios en cada procedimiento.

TURTLE: Líneas (9000-9020) No necesita parámetros.

FORWARD: Líneas (9Ø3Ø-9Ø9Ø) Parámetro L.

PENDOWN: Líneas (91ØØ-911Ø) No tiene parámetros, si quiere cambiar la tinta bastará con hacer INK tinta.

PENUP: Líneas (912Ø-913Ø) No necesita parámetros.

SETHD: Líneas (914Ø-915Ø) Parámetro A. Ej. LET A=9Ø: GO SUB SETHD.

SETXY: Líneas (9160-9170) Parámetros X e Y.

BACK: Líneas (9180-9190) Parámetro L.

TURN: Líneas (9200-9210) Parámetro A.

VECTOR: Líneas (922Ø-923Ø) Parámetros A y L.

RIGHT: Líneas (924Ø-925Ø) Parámetro A.

LEFT: Líneas (926Ø-927Ø) Parámetro A.

ARCR: Líneas (928Ø-932Ø) Parámetro R– R y GR.

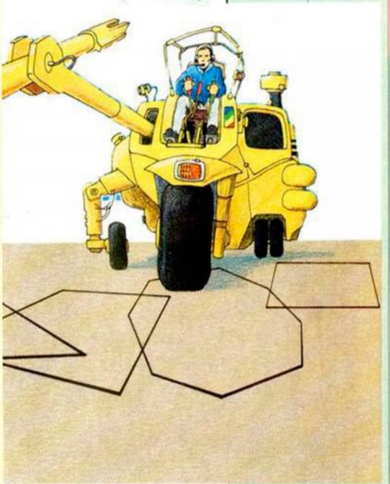
Cuando empleemos esta subrutina en nuestros programas en Basic debemos poner como primera instrucción antes de emplear cualquiera de la subrutina que forman el TURTLE, GO SUB 9400 esta subrutina inicializa todas las variables para poder trabajar.

A continuación aparece el listado en Basic, y un pequeño programa de aplicación que introduce una técnica muy potente y desconocida por muchos, que es la recursividad (Prog. 2).



PROGRAMA 2

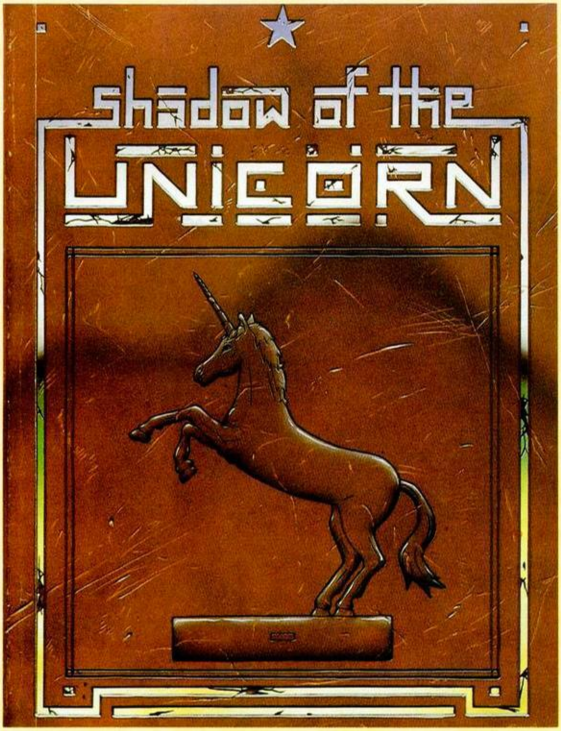
```
   5 REM ** ESPIRALES **
  10 GO SUB 9400
  20 GO SUB TURTLE
  30 LET L=1: LET A=60: GO SUB 1
00: STOP
 100 GO SUB FORWARD: GO SUB RIGH
T: LET L=L+1: LET A=60: GO SUB 1
00: RETURN
9000 REM ** turtle **
9010 CLS : LET xcor=127: LET yco
r=87
9020 PAPER 7: INK 0: RETURN
9030 REM ** forward **
9040 PLOT INT (xcor+0.5),INT (yc
or+0.5)
9050 LET newx=xcor+l*COS (headin
g*PI/180)
9060 LET newy=ycor+l*SIN (headin
g*PI/180)
9070 DRAW INT (newx+0.5)-INT (xc
or+0.5),INT (newy+0.5)-INT (ycor
+0.5)
9080 LET xcor=newx: LET ycor=new
y
9090 RETURN
9100 REM **penon**
9110 INVERSE 0: RETURN
9120 REM **penoff**
9130 INVERSE 1: RETURN
9140 REM **sethd**
9150 LET heading=a: RETURN
9160 REM **setxy**
9170 LET xcor=x: LET ycor=y: RET
URN
9180 REM **back**
9190 LET l=-l: GO SUB forward: R
ETURN
9200 REM **turn**
9210 LET heading=heading+a: RETU
RN
9220 REM **vector**
9230 GO SUB sethd: GO SUB forwar
d: RETURN
9240 REM **right**
9250 LET a=-a: GO SUB turn: RETU
RN
9260 REM **left**
9270 GO SUB turn: RETURN
9280 REM **arcr**
9290 FOR k=1 TO gr
9300 LET l=r: GO SUB forward
9310 LET a=1: GO SUB turn
9320 NEXT k: RETURN
9330 REM **punto**
9340 GO SUB setxy
9350 PLOT xcor,ycor: RETURN
9400 REM **inicializacion**
9410 LET turtle=9000: LET forwar
d=9030: LET penon=9100: LET peno
ff=9120: LET sethd=9150: LET set
xy=9160: LET back=9180: LET turn
=9200: LET vector=9220: LET righ
t=9240: LET left=9260: LET arcr=
9280: LET punto=9330
9420 LET heading=0: LET a=0: LET
 l=0: LET r=0: LET gr=0: LET x=0
: LET y=0: LET xcor=0: LET ycor=
0: RETURN
```



PROGRAMA 1

```
   1 PROGRAM CUADRADO;
   2 VAR
1360 PROCEDURE GIROCUADRADO (LADO:INTEGER);
1370 VAR
1380 J,I:INTEGER;
1390 BEGIN (*COMIENZA*)
1400 FOR I:=1 TO 12 DO
1410 BEGIN (* 12 puntos *)
1420 SETXY (127,60);(*Posicion de giro*)
1430 FOR J:=1 TO 4 DO
1440 BEGIN (* PINTO CUADRADO *)
1450 FWD (LADO);
1460 RIGHT (90)
1470 END;(* FIN PINTAR CUADRADO *)
1480 TURN (30)(* GIRAMOS 30 GRADOS PARA DIBUJAR SIGUIENTE CUADRADO *)
1490 END (* 12 puntos *)
1500 END;(* FIN PROCEDIMIENTO *)
1510 BEGIN (* COMIENZA PROGRAMA *)
1520 TURTLE;(* INICIALIZO *)
1530 SETXY (127,60);(* FIJO POSICION *)
1540 SETHD (0);(* FIJO POSICION *)
1550 GIROCUADRADO (30);(* Llamada al procedimiento para dibujar cuadrados *)
1560 SETXY (127,60);SETHD (90)
1570 PENUP; (* LEVANTO PLUMA *)
1580 BACK (30);(* BAJO LA LONGITUD 30 *)
1590 PENDOWN (7);(* LEVANTO LAPIZ CON TINTA BLANCA *)
1600 SETHD (0);(* FIJO DIRECCION *)
1610 ARCR (0.527,360) (* UNO PUNTOS PARA formar CIRCUNFERENCIA *)
1620 END
```

SI BUSCAS LO MEJOR



SHADOW OF THE UNICORN

UNA NUEVA ERA PARA TU SPECTRUM. UN PROGRAMA QUE INCORPORA UN INTERFACE AMPLIANDO LA MEMORIA A 64 K. CON SALIDA PARA JOYSTICK Y QUE TIENE ¡6.800 PANTALLAS! PUDIENDO CONTROLAR HASTA 10 PERSONAJES DIFERENTES.

¡UNA AUTENTICA REVOLUCION!

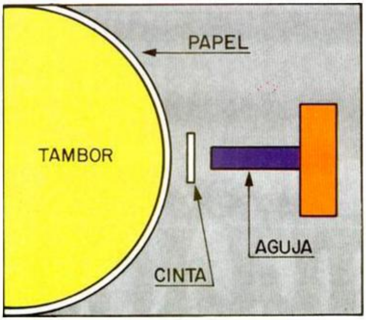


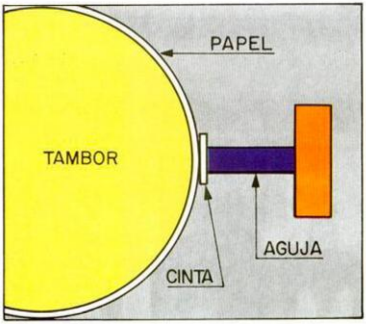
CRITICAL

EL NUEVO Y ESPERADO JUEGO DE LANZO "SCUBA DIVE", "HARRIER" ENTRE OTROS.

UN JUEGO DE ACCION A TOPE EN TODA TU HABILIDAD.

DISTRIBUCION EXCLUSIVA PAR
SANTA ENGRACIA, 17 - TEL.





Una impresora de impacto genera una imagen, por matriz de puntos, cuando las diminutas agujas de la cabeza de impresión golpean la cinta entintada colocada entre el papel y la cabeza.

Especial impresoras

Toda persona metida en el mundo de la informática conoce o ha oído hablar de una impresora, pero cuál es la idónea para su aplicación particular, qué tipo de Interface necesita o simplemente ¿cómo hacerla funcionar? En este artículo se ofrece al lector una GUIA con las características más importantes de las impresoras que nos ofrece el mercado nacional, así como unos conocimientos técnicos sobre este extendido dispositivo.

Cada día está más extendido el uso de las impresoras y plotter, ya sea para aplicaciones de estadística, depuración de programas, Hard Copy o simplemente como herramienta imprescindible de un procesador de textos.

No hace muchos años las impresoras eran unos periféricos que los asociábamos a grandes ordenadores y que sólo se encontraban en Centros de Cálculo y en Laboratorios de I + D (Investigación y Desarrollo); hoy gracias al avance de la tecnología, que reduce los costos de fabricación, podemos encontrarlos en los sitios más dispares, desde los grandes almacenes donde nos anotan el total de las compras efectuadas hasta en las farmacias, donde instrumentos de electromedicina imprimen los datos referentes a nuestra tensión arterial.

Los usuarios de ordenadores personales, podemos acceder hoy en día a estos periféricos, que hace tiempo nos estaban vetados por su elevado costo. En las tiendas de informática se nos ofrecen una gran variedad de impresoras, en cuanto a tamaño y precio se refiere, por tanto, con ayuda de esta «guía», podremos elegir aquélla que dentro de nuestras posibilidades económicas se adapte mejor a nuestras necesidades de trabajo.

Diversos tipos

Las impresoras pueden clasificarse, atendiendo a su funcionamiento básico, en diversos tipos:

- MATRICIALES
- MARGARITA
- TERMICAS
- ELECTROSTATICAS
- INYECCION DE TINTA
- LASER

Sin duda alguna las impresoras más populares, entre los usuarios de ordenadores, son las *matriciales*, ya que ofrecen unas características de calidad de impresión y velocidad que satisfacen cualquier necesidad; además su costo es relativamente bajo en comparación con los otros tipos, aunque todo depende de lo sofisticado que sea el modelo elegido.

Su funcionamiento básico es muy simple; constan de una cabeza que se desplaza de izquierda a derecha a lo ancho del papel, en cuyo interior se encuentran unas bobinas, conocidas por el nombre de *solenoides*, que alojan cada una de ellas una diminuta aguja sin punta; cuando la bobina se activa, mediante la aplicación de un pulso eléctrico, la aguja sale de su alojamiento y golpea una cinta entintada colocada entre el papel y la cabeza de impresión; cuando el pulso desaparece, la aguja vuelve a su posición de reposo. En el sitio donde la aguja golpea al papel queda marcado un punto.

Disponiendo las agujas, generalmente nueve, de manera que formen una columna y activándolas y desplazando la cabeza de impresión convenientemente, se pueden componer todo tipo de caracteres que estén comprendidos dentro de una matriz imaginaria de 9 x 7, 9 x 9, etc., puntos, dependiendo de las características de la impresora.

Todo este funcionamiento es *trasnsparente* para el usuario ya que los circuitos internos se encargan, una vez recibido el carácter a imprimir, de seleccionar aquellas agujas que lo formen de acuerdo con un generador interno de caracteres. Debido a su funcionamiento también son conocidos como *Impresoras de impacto por matriz de puntos.*

Lógicamente los caracteres no estarán formados por un trazo continuo, si no por diminutos puntos separados entre si. Algunas impresoras incorporan un sistema que permite rellenar esos huecos con una segunda pasada.

Para aumentar la veloci-

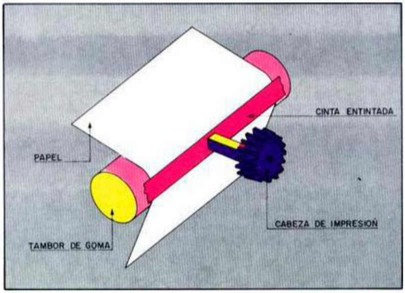
dad de impresión hay un sistema denominado *bidireccional* que imprime también cuando el cabezal retorna al margen izquierdo. Esta facilidad puede ser inhibida cuando se realizan gráficos de calidad y se desea una correcta correspondencia entre las columnas de una y otra pasada.

Para la realización de gráficos de alta resolución, la impresora debe permitir que el usuario, con un software adecuado, desactive el juego de caracteres interno y controle cada aguja por separado.

El papel que utilizan es diverso, dependiendo del sistema que tengan de arrastre: tracción o fricción. Para el primero se requiere papel perforado en sus extremos, también se denomina *continuo* ya que las hojas van unidas y plegadas por una línea de puntos para facilitar su separación. En los sistemas de fricción pueden utilizarse rollos de papel similares a los utilizados en las máquinas registradoras o utilizar hojas sueltas tipo folio; para los primeros se requieren unos soportes especiales y para las hojas, si no se quieren introducir una a una, existen unos *alimentadores de papel*.

Las impresoras de *margarita* tienen una calidad de impresión comparable a la de una máquina de escribir. En cada brazo de la margarita, por su similitud con esta flor, hay grabado un carácter en relieve; para imprimir debe posicionarse el brazo elegido frente a una especie de martillo que lo golpea contra una cinta especial colocada entre el papel y la margarita. El carácter se imprime de una sola vez y con trazo continuo, sin embargo su velocidad es lenta debido a su mecanismo de selección que debe de girar una y otra vez hasta encontrar el carácter elegido.

Las impresoras *térmicas* sustituyen las agujas de impresión por diminutas *resistencias eléctricas* que tienen



la facilidad de calentarse y enfriarse rápidamente. El papel utilizado es especial, ya que cambia de tono al recibir el calor de la resistencia.

Las impresoras *electrostáticas* utilizan unas cabezas de impresión que generan pequeñas descargas eléctricas, al tratar el papel especial sensibilizado, con un *toner*, cambia el color en aquellos puntos donde recibió la descarga.

Tanto las impresoras térmicas como las electrostáticas tienen una calidad de impresión idéntica a las de impacto.

Las impresoras de *inyección de tinta* y las de *láser* están pensadas para operar con ordenadores más potentes, donde el volumen de impresión es grande; generalmente se encuentran fuera del alcance del usuario medio ya que tienen un elevado precio.

Plotter

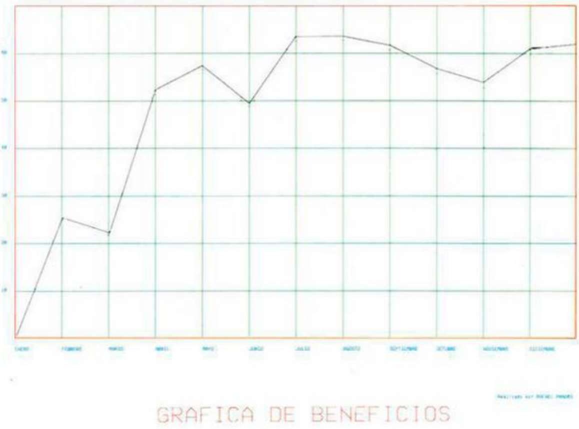
Con el término *plotter* se denominan ciertos periféricos con los que se puede realizar cualquier tipo de dibujo, desde los típicos gráficos de barras hasta los más sofisticados planos de piezas o edificios, con la misma calidad que el dibujado por un experto delineante.

De los diversos tipos existentes, los más utilizados con ordenadores personales son los *plotter de pluma*, donde el elemento trazador de los gráficos son unos diminutos bolígrafos de colores que pueden seleccionarse a voluntad.

El modo de funcionamiento varía de un modelo a otro, pero el más generalizado de los denominados plotter de pluma *tipo plano*, consiste en una cabeza portaplumas que puede desplazarse, mediante una guía, de derecha a izquierda (coordenada X); el papel que va aprisionado entre dos rodillos se desplaza hacia arriba y hacia abajo (coordenada Y). Mediante el correcto de estos dos movimientos (X e Y) se puede realizar cualquier tipo de dibujo.

Entre las funciones que se pueden programar a través de comandos "LPRINT", se encuentran:

– Establecer la posición actual de la pluma como origen X-Y.



— Seleccionar el color de la pluma.

— Seleccionar el trazado de línea (continuo o discontinuo).

— Mover la pluma, sin trazar nada, hasta las coordenadas X e Y.

— Trazar una línea desde la posición actual de la pluma hasta las coordenadas X1 e Y1, después hasta X2 e Y2, etc.

— Movimientos y trazados relativos.

— Desplazar la pluma hasta el origen.

— Imprimir caracteres.

— Seleccionar el tamaño de los caracteres.

— Seleccionar la dirección de impresión de los caracteres (arriba, abajo, derecha o izquierda).

— Mover la pluma al comienzo de la siguiente línea.

Sencillas de manejar

Las impresoras son unos periféricos extremadamente sencillos de manejar y por tanto no requieren unos conocimientos profundos de informática.

La conexión eléctrica se realiza normalmente a través de un *cable cinta* que conexiona la impresora directamente con el ordenador o como ocurre en la mayoría de los casos a un interface apropiado, el cual deberá conectarse al ordenador; esta operación deberá realizarse con la alimentación de los dispositivos desconectados.

Cualquier impresora de calidad media dispone de varios pulsadores, incluido el de encendido ON/OFF, y de una serie de LED o indicadores luminosos que nos muestran los diversos estados de funcionamiento. La misión de estos pulsadores es la siguiente:

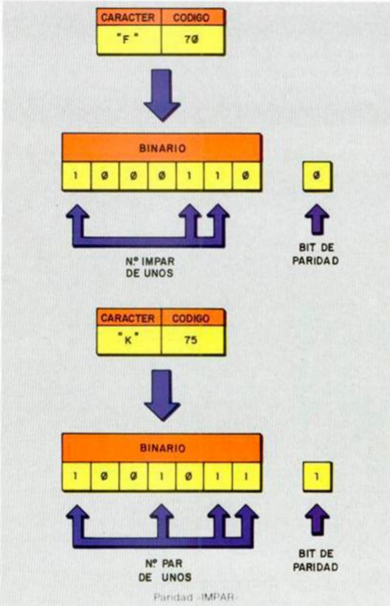
SEL. Selecciona alternativamente la habilitación o inhibición de la transmisión de datos; generalmente lleva asociado un indicador luminoso que permite conocer el estado de disponibilidad de impresión. Cuando la impresora está habilitada se pueden ejecutar desde el ordenador cualquiera de los comandos "LPRINT", "LLIST", o "COPY"; cualquier duda sobre el manejo de estas sentencias las podrá resolver consultando el capítulo "Gestión de impresora", del curso de BASIC publicado en esta revista.

LF. Cada vez que se pulsa, el papel avanza una línea, quedando, por tanto, posicionada la cabeza de impresión en la línea siguiente.

TOF o *FF.* Este pulsador permite desplazar el papel hasta el comienzo de la siguiente página.

Estas dos últimas opciones, solamente son operativas cuando la selección está inhibida.

Otro indicador luminoso es el denominador PE (paper end) que nos avisa cuando se termina el papel.



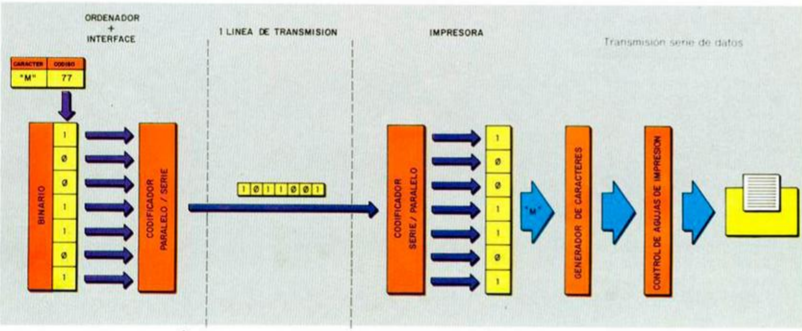
Paridad IMPAR

Estas siglas aunque adoptadas por la mayoría de las impresoras puede que no coincidan con la que usted vaya a manejar.

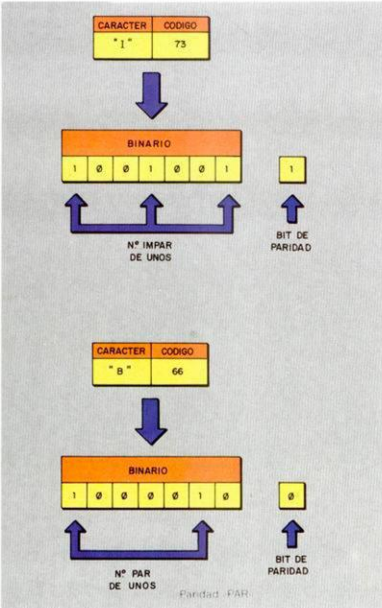
Otra función que suelen incorporar es la de TEST, mediante la cual se imprime un "patrón" que permite determinar si las agujas funcionan correctamente.

Necesidad de un interface

Cuando se desea acoplar al Spectrum una impresora con más de 32 columnas, con una velocidad de impresión alta, que tenga varios



Transmisión serie de datos



Paridad PAR

juegos de caracteres y que además se pueda programar el tamaño de los caracteres, por ejemplo, nos encontramos ante la imposibilidad de poderla conectar directamente.

Todo el problema viene originado por que la mayoría de las impresoras están diseñadas para operar con unos protocolos de comunicación distintos del que utiliza el Spectrum, que es muy específico y particular.

Para solucionar esta incompatibilidad, es necesario conectar un dispositivo, conocido por el nombre "Interface", entre el ordenador y la impresora. Este interface permite adaptar los distintos protocolos para que exista una comunicación o diálogo entre ambos.

Entre los diversos protocolos de comunicación, los más utilizados por estos periféricos son los denominados "CENTRONICS" y "RS-232C".

El primero se basa en la transmisión paralela de datos y el segundo, por el contrario, en la transmisión serie.

A la hora de comprar una impresora deberá tener en cuenta el tipo de interface que deberá utilizar para que ésta funcione correctamente. En los comercios especializados de informática, se pueden encontrar interfaces que adaptan el Spectrum a cualquiera de estos dos protocolos.

Deberá leer atentamente las instrucciones del manual del interface que maneje, que le indicarán la forma de operar con él.

Dependiendo del tipo de interface e impresora elegida, la sentencia "COPY", no podrá ser ejecutada como de costumbre; en la mayoría de los casos deberá ejecutar previamente un pequeño programa en C/M (código máquina) bien residente en una memoria tipo ROM alojada en el interface (firmware) o bien grabada en una cinta de cassette.

Transmisión de datos

En la transmisión de datos paralela, se utiliza una línea (cable) por cada bit que envía, por tanto, se necesitarían siete líneas; sin embargo, en la transmisión serie, se utiliza una única línea para transmitir los siete bits.

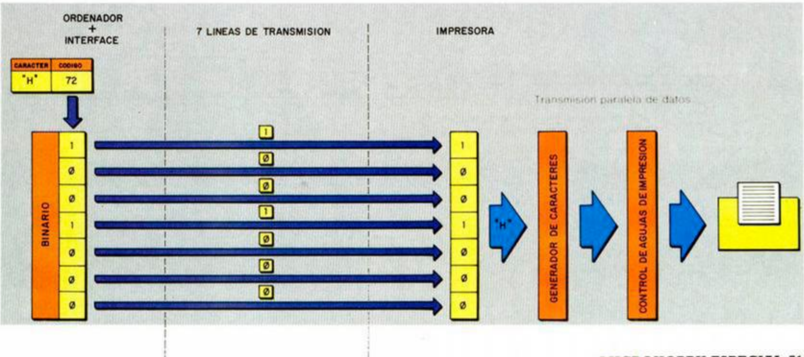
Con la transmisión paralela se puede alcanzar mayor velocidad de transmisión, expresada en *baudios*, ya que en la impresora se reciben todos los bits a la vez y en la transmisión serie es necesario efectuar siete lecturas para reconocer el carácter que le llega.

A pesar de tener una velocidad de transmisión más baja, el protocolo serie se utiliza frecuentemente en conexiones relativamente largas, ya que es menor el número de cables que se necesitan. De los dos tipos de transmisión serie que existen, síncrona y asíncrona, la más utilizada es la última.

En la transmisión serie de tipo asíncrona, cuando el interface quiere enviar un carácter, debe transmitir primeramente un bit de comienzo (START), a continuación los siete bits del código ASCII a transmitir, luego un bit de *paridad* (control) y por último uno o dos bits de parada (STOP).

El bit de paridad se utiliza para detectar errores en la transmisión de datos. Existen dos tipos de paridad: *Par* e *Impar*.

Cuando se utiliza el proto-



Transmisión paralela de datos



A L I E N
P.V.P.
2.495
Ptas.
En el espacio nadie oirá tus gritos.
NAVEGANTE
Tímida, hábil e inteligente, asustadiza
OFICIAL EJECUTIVO
Directo, imaginativo, cauteloso y leal
OFICIAL CIENTIFICO
Reservado, antipático; brillante, ocasionalmente ilógico
CAPITAN
Sólido, digno de confianza, valeroso, excelente líder
OFICIAL DE MAQUINAS
Físicamente fuerte, bajo índice de inteligencia, rebelde en potencia
CONTRAMAESTRE
Testaruda, ambiciosa, autoritaria, fértil en recursos
OFICIAL DE MAQUINAS
Cínico, rebelde, indigno de confianza, imperturbable
El persona
a bordo
ejecutará
tus órdene
casi
siempre . .
MIND GAMES ESPAÑA, S.A.
MGE SOFTWARE
SPECTRUM 48K · CBM64
Empleando el
genial sistema de
Control de Personalidad
Argus Press
Software Group
MIND GAMES ESPAÑA, S.A. · Mariano Cubí, 4 entlo. Tel. 218 34 00 · 08006 Barcelona (España)

colo con paridad par, el interface pone a "1" este bit si el número de "unos" del carácter a enviar es impar, y por el contrario lo pone a "Ø" cuando hay un número par de "unos".

El protocolo de paridad impar funciona a la inversa, es decir, pone el bit, de paridad a "1" cuando el número de "unos" es par, y a "Ø" cuando es impar. De esta manera la impresora puede detectar si hay algún error en la transmisión de datos, contando el número de bits a "uno" que recibe por cada carácter.

Como no todos los interfaces utilizan las mismas velocidades de transmisión, ni la misma cantidad de bits dentro del protocolo serie, algunas impresoras, generalmente las de cierta calidad, incorporan unos microinterruptores (switches) encapsulados en formato DIP (terminales en doble línea) que permiten seleccionar algunas de estas características:

- Velocidad en baudios.
- Bit de paridad (SI/NO).
- Tipo de paridad (PAR/IMPAR).
- N.º de bits a transmitir (7/8).
- N.º de bits de parada (1/2).

Estos microinterruptores suelen estar alojados en el interior de las impresoras, por lo que es necesario desmontar la cubierta; aunque en algunos modelos están accesibles desde el exterior, generalmente, la selección debe efectuarse antes de conectar la impresora, ya que es precisamente en el momento del encendido cuando la circuitería interna efectúa la lectura del estado (ON/OFF) de los microinterruptores.

Buffer de impresión

Dentro de la memoria RAM o de usuario, hay reservados 256 bytes que el S.O. (sistema operativo) utiliza como buffer intermedio de impresión. La dirección de comienzo se encuentra en la posición 23296 y la última en la 23551, ambas expresadas en decimal.

Cuando se desea imprimir a través de los comandos "LLIST", "LPRINT" o "COPY" se vuelcan en este buffer los datos que van a ser transmitidos a la impresora; una vez realizada esta tarea dicho área se resetea (contenido cero).

En este buffer se pueden almacenar como máximo 32 caracteres de 8 bytes (32 x 8 = 256 bytes), es decir una fila de la pantalla del T.V. o monitor. La información está estructurada de la siguiente manera:

Los 32 primeros bytes contienen los datos correspondientes al primer byte de cada carácter, los 32 siguientes al byte segundo, y así sucesivamente hasta el octavo byte; de manera que la información perteneciente al primer carácter estará distribuida en las siguientes posiciones y direcciones

POSICION RELATIVA	DIRECCION ABSOLUTA
1	23296
33	23328
65	2336Ø
97	23392
129	23424
161	23456
193	23488
225	2352Ø

Las impresoras más sencillas, es decir, aquellas que no requieren de un interface y que se conexionan directamente al conector de expansión, utilizan este buffer intermedio.

El programa n.º 1 permite introducir un carácter o cadena en dicha zona y poder visualizar el contenido de sus posiciones antes de que se reseteen.

Las impresoras de calidad disponen de su propio buffer de impresión. El interface va almacenando la información a una velocidad de transmisión relativamente alta, en dicha memoria, y la impresora la va recogiendo e imprimiendo a una velocidad inferior. Cuando el ordenador ha terminado de enviar toda la información, queda libre y puede realizar otras tareas, mientras tanto la impresora termina su labor.

Códigos de control

Dentro del código ASCII convencional existe un carácter denominado "ESCAPE" que es utilizado por la mayoría de las impresoras para seleccionar por software algunas de sus funciones especiales, como por ejemplo:

- Color de impresión.
- Tipo de letra.
- Tamaño de la misma.
- Juego de caracteres.
- Etc...

Este carácter, cuyas siglas son "ESC", tiene el código decimal "27" y en hexadecimal el "1B". La forma de enviarlo es utilizando la sentencia "LPRINT":

LPRINT CHR$ 27

Al detectar la impresora este código, se prepara para recibir una secuencia de caracteres alfanumérica que para ella tienen un significado y que se corresponden con algunas de sus funciones especiales; veamos unos ejemplos:

FUNCIONES	COMANDOS
Símbolos gráficos	ESC#
20 cpi	ESC[3w
Comienzo subrayado	ESC x
Color rojo	ESC[Øt

La forma de enviarlos sería:

LPRINT CHR$ 27 + "#"
LPRINT CHR$ 27 + "[3w"
LPRINT CHR$ 27 + "x"
LPRINT CHR$ 27 + "[Øt"

Para manejar con «secuencias ESCAPE" su impresora deberá leer atentamente las instrucciones del manual.

Otros códigos ASCII, utilizados con frecuencia en el manejo de impresoras, son:

LINE FEE. Avanza el papel a la siguiente línea.

FORM FEED. Avanza el papel hasta el principio de la siguiente página.

CARRIAGE RETURN. Retorna la cabeza de impresión al principio de la línea. En algunas impresoras se puede seleccionar que al recibir este código se efectúe también un Line Feed.

SHIFT OUT. Permite la impresión en doble ancho.

SHIFT IN. Inhibe la característica anterior.

CODIGO DE CONTROL	SIGLAS	CODIGO DECIMAL	CODIGO HEXA.
LINE FEED	LF	1Ø	ØA
FORM FEED	FF	12	ØC
CARRIAGE RETURN	CR	13	ØD
SHIFT OUT	SO	14	ØE
SHIFT IN	SI	15	ØF
ESCAPE	ESC	27	1B

TABLA I
Códigos "ASCII" de control utilizados frecuentemente en el manejo de impresoras.

Es usted capaz de tomar el relevo del General Montgomery...?

& Juegos ESTRATEGIA

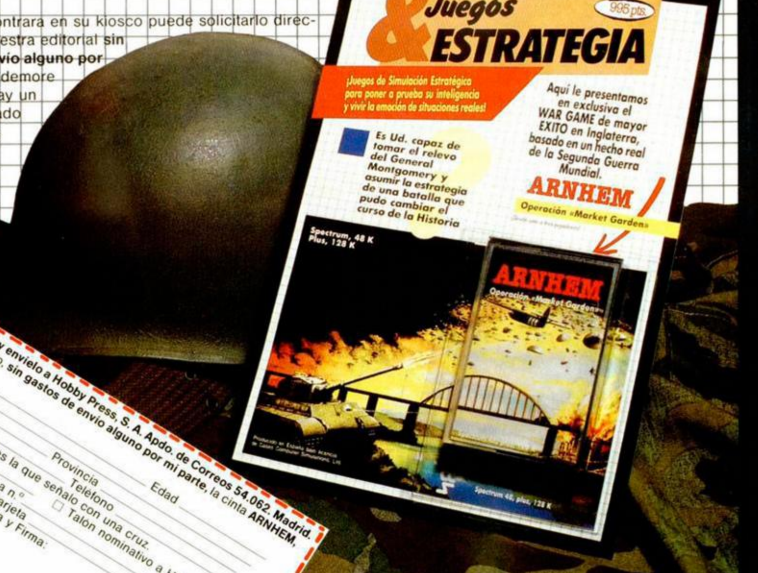
Ya está a la venta!

le presenta en exclusiva el WAR GAME, para Spectrum, de mayor éxito en Inglaterra:

ARNHEM

(operación «*Market Garden*», basada en un hecho real de la Segunda Guerra Mundial)

Si no lo encontrara en su kiosco puede solicitarlo directamente a nuestra editorial **sin gastos de envío alguno por su parte.** No demore su pedido, hay un número limitado de cassettes.



Recorte o copie este cupón y envíelo a Hobby Press, S. A. Apdo. de Correos 54.062. Madrid.
Deseo recibir en mi domicilio, **sin gastos de envío alguno por mi parte,** la cinta **ARNHEM,** al precio de 995 pesetas.
Nombre ______ Edad ______
Dirección ______
Localidad ______ Provincia ______
Código ______ Teléfono ______
La forma de pago elegida es la que señalo con una cruz.
☐ Giro Postal n.º ______ ☐ Tarjeta Visa n.º ______ ☐ Talón nominativo a Hobby Press, S.A.
Fecha de caducidad de la tarjeta ______
Fecha y Firma:

HOBBY PRESS, S.A. Editamos para gente inquieta.

Guía de impresoras

TERMINOLOGIA

CPS: Unidad para expresar la velocidad de impresión en "caracteres por segundo".

LPM: Idem "líneas por minuto".

N.º COLUMNAS: Número de caracteres que es capaz de imprimir en una línea.

CPI: Cantidad de "caracteres por pulgada"; determinan el ancho de la letra.

LPI: Cantidad de "líneas por pulgada"; determinan la densidad de impresión.

INCH: Medida de longitud inglesa (pulgada) equivalente a 2,5401 centímetros. Se utiliza frecuentemente para expresar la anchura de los caracteres o del papel.

BPS: Velocidad de transmisión expresada en "bits por segundo".

BAUDIO: Unidad de velocidad de transmisión. 1 baudio = 1 bps.

CENTRONICS: Protocolo paralelo de comunicación.

RS-232C: Protocolo serie de comunicación.

PARITY EVEN: Control de errores de transmisión basado en el chequeo de "paridad par".

PARITY ODD: Idem "paridad impar".

DOT MATRIX: Sistema de impresión por matriz de puntos.

PRINT HEAD: Cabeza de impresión donde se alojan, por ejemplo, las agujas de una impresora de impacto.

BIT IMAGE: Sistema incorporado en algunas impresoras, mediante el cual se pueden imprimir gráficos en alta resolución, ya que permiten controlar por software cada aguja independientemente.

HARD COPY: Término referente a la conexión de una impresora a un ordenador, de manera que ésta imprima todo lo que se visualiza en la pantalla del monitor o T.V.



Marca: Seikosha
Modelo: GP-50 S

Otras características: Permite utilizar el comando "COPY"
Accesorios: Se acompaña con alimentador externo
Precio aproximado: 19.990 pesetas

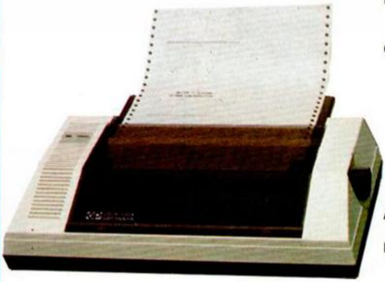
Columnas: 32
Velocidad: 35 cps
Protocolo: Compatible Spectrum
Arrastre: Fricción
Papel: Rollo (ancho máx. 5 pulgadas)



Marca: Seikosha
Modelo: GP-700 A (color)

Columnas: 80-106
Velocidad: 50 cps
Protocolo: Centronics
Arrastre: Fricción/Tracción
Papel: Papel continuo y folios sueltos de 10 pulgadas
Otras características:
- Impresora de color (negro, magenta, rojo, amarillo, verde, cyan y azul)
- Caracteres normales y expandidos
- Modalidad gráfica
- Cuatro tipos de caracteres

Accesorios: Interface opcional RS-232C
Precio aproximado: 69.900 pesetas

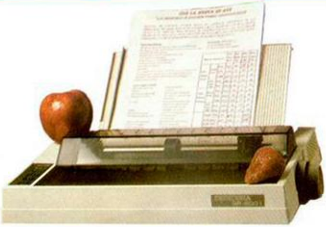


Marca: Seikosha
Modelo: GP-500 AS

Columnas: 80
Velocidad: 50 cps
Protocolo: RS-232C
Arrastre: Tracción
Papel: Continuo
Otras características:
- Caracteres normales y expandidos. Gráfica.
- Conexión directa al interface-1 del Spectrum

Precio aproximado: 49.900 pesetas

s de impacto por matriz de punto



Marca: Seikosha
Modelo: SP-800

Columnas: 80-137
Velocidad: 96 cps
Protocolo: Centronics
Arrastre: Fricción/Tracción
Otras características:
- Bidireccional
- Volcado de datos en hexadecimal (DUMP)
- Introductor automático hoja a hoja
- Letra de calidad (24 cps)
- Fijación de márgenes
- Más de 200 combinaciones de caracteres
- Capacidad gráfica de alta resolución

Accesorios: Interface opcional serial
Precio aproximado: 59.900 pesetas



Marca: Seikosha
Modelo: SP-1000 A

Columnas: 80-137
Velocidad: 100 cps
Protocolo: Centronics
Arrastre: Fricción/Tracción
Otras características:
- Las mismas que la SP-800
- Juego de caracteres controlados por secuencias ESCAPE
- 96 caracteres programables por el usuario
- La impresora SP-1000 AS tiene las mismas características pero con protocolo RS-232C

Precio aproximado: 69.900 pesetas



Marca: Seikosha
Modelo: BP-5200 A

Columnas: 136
Velocidad: 200 cps
Protocolo: Centronics y RS-232C
Buffer: 4 Kb ampliables a 20 Kb
Arrastre: Fricción/Tracción
Papel: Desde 5 hasta 15,5 pulgadas
Otras características:
- Letra de calidad (106 cps)
- Caracteres normales y expandidos
- Capacidad gráfica de alta resolución
- Más de 150 tipos de letra
- Bidireccional

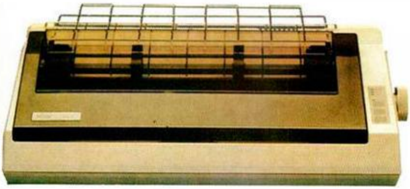
Accesorios: Introductor automático de documentos opcional
Precio aproximado: 199.000 pesetas

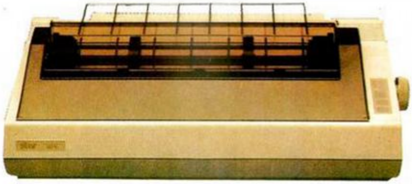


Marca: Seikosha
Modelo: BP-5420 A

Columnas: 136
Velocidad: 420 cps
Protocolo: Centronics y RS-232C
Buffer: 18 Kb
Arrastre: Fricción/Tracción
Papel: Desde 5 hasta 15,5 pulgadas
Otras características:
- Tipo de letra seleccionable por HARD y SOFT
- Las mismas que la BP-5200 A

Precio aproximado: 299.000 pesetas





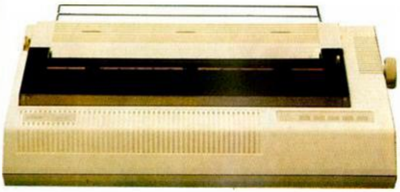
Marca: Star
Modelo: SG 10/15

Columnas: 80-136 (SG 10), 136-233 (SG 15)
Velocidad: 120 cps
Protocolo: Centronics
Buffer: 2 Kb expansibles a 6 Kb (SG 10), 16 Kb (SG 15)
Arrastre: Fricción/Tracción
Tinta: Standard máquina de escribir
Otras características:
- Bidireccional
- Letra de calidad "NLQ" (40 cps)
- Gráficos de alta resolución (bit image)
- 240 caracteres definidos
- Tabuladores horizontales y verticales

Accesorios: Interface opcional RS-232C
Precio aproximado: 79.500 pesetas (SG 10) 109.500 pesetas (SG 15)

Marca: Star
Modelo: SD 10/15

Columnas: 80-136 (SD 10) 136-233 (SD 15)
Velocidad: 120 cps
Protocolo: Centronics
Buffer: 2 Kb expansibles a 6 Kb (SD 10), 16 Kb (SD 15)
Arrastre: Fricción/Tracción
Otras características: Las mismas que la SG 10/15
Accesorios: Interface opcional RS-232/20 mA por bucle de corriente
Precio aproximado: 109.500 pesetas (SD 10) 139.500 pesetas (SD 15)





Marca: Riteman
Modelo: F+

Columnas: 80
Velocidad: 105 y 52 cps
Protocolo: Centronics
Buffer: 2 Kb
Arrastre: Fricción/Tracción
Papel: Entre 4 y 10 pulgadas
Otras características:
- Correcta impresión de los trazos descendentes: j, g, p, q
- 32 caracteres definibles por usuario (con 2 Kb de buffer)
- 256 caracteres definibles por usuario (con 8 Kb de buffer)
- Plotter gráfico de 576 puntos
- Letra de calidad "NLQ"

Accesorios:
- Interface opcional RS-232C
- Ampliación a 8 Kb de memoria

Precio aproximado: 69.000 pesetas

Marca: Star
Modelo: SR 10/15

Columnas: 80-136 (SR 10) 136-233 (SR 15)
Velocidad: 200 cps
Protocolo: Centronics
Buffer: 2 Kb expandibles a 6 Kb (SR 10), 16 Kb (SR 15)
Arrastre: Fricción/Tracción
Otras características:
- Las mismas que la SD 10/15
- Inserción automática de hojas
- Corte de hojas a ras del papel

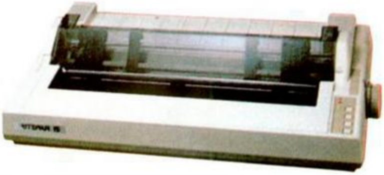
Accesorios: Interface opcional RS-232C
Precio aproximado: 154.500 pesetas (SR 10) 189.500 pesetas (SR 15)



Marca: Riteman
Modelo: 10

Columnas: 80
Velocidad: 120 cps
Protocolo: Centronics
Buffer: 2 Kb
Arrastre: Fricción/Tracción
Accesorios:
- Juego de EPROM's "NLQ" (opcional)
- Interface opcional RS-232C
- Tractor de arrastre opcional

Precio aproximado:
81.000 pesetas



Marca: Riteman
Modelo: 15

Columnas: 136
Velocidad: 160 cps
Protocolo: Centronics
Buffer: 2 Kb
Arrastre: Fricción/Tracción
Papel: 15 pulgadas
Otras características:
- 128 combinaciones de escritura distintas
- Alimentador de papel por tractor incorporado
- No es necesaria su desconexión para ajustar el formato

Accesorios:
- Juego de EPROM's "NLQ" (opcional)
- Interface opcional RS-232C

Precio aproximado:
155.000 pesetas



Marca: Newprint
Modelo: DP-100

Columnas: 80-142
Velocidad: 100 cps
Protocolo: Centronics o RS-232C
Arrastre: Fricción/Tracción
Papel: Entre 101,6 y 254 mm.
Otras características:
- Bidireccional
- Posibilidad gráfica (bit image)
- Caracteres normales y expandidos
- Códigos de control por software

Precio aproximado:
Con interface paralelo 53.900 pesetas
Con interface serie 59.900 pesetas



Marca: C. Itoh
Modelo: 7500 AP/AR

Columnas: 80-136
Velocidad: 105 cps
Protocolo: Centronics o RS-232C
Buffer: 2 Kb o 16 Kb
Arrastre: Fricción/Tracción
Papel: 4 a 10 pulgadas
Otras características:
- Posibilidad gráfica
- Diversos juegos de caracteres

Precio aproximado:
Con interface paralelo 81.000 pesetas
Con interface serie 89.000 pesetas



Marca: C. Itoh
Modelo: 8510 BP/BC2

Columnas: 80-136
Velocidad: 120 cps
Protocolo: Centronics o RS-232C y 20 mA
Buffer: 2 Kb
Arrastre: Fricción/Tracción
Otras características:
- Diversos tipos de caracteres
- Posibilidad gráfica (bit image)
- Diversas velocidades de transmisión. Bidireccional
- Selección del control de paridad (par/impar)

Accesorios: Alimentador de hojas
Precio aproximado:
Con interface paralelo 104.900 pesetas
Con interface serie 119.900 pesetas
Nota: Las impresoras 8510 SP y SR constituyen una versión mejorada.
- Velocidad 180 cps
- Posibilidad de generar caracteres especiales
- Mayor buffer de memoria
- Volcado hexadecimal
- La serie SC incorpora la opción de color
- Precio:
  Con interface paralelo 119.900 pesetas
  Con interface serie 134.900 pesetas



Marca: C. Itoh
Modelo: 1550 BP/BC2

Columnas: 136-230
Velocidad: 120 cps
Protocolo: Centronics o RS-232C y 20 mA
Buffer: 2 Kb
Arrastre: Fricción/Tracción
Papel: Entre 113 y 394 mm.
Otras características:
- Bidireccional
- Posibilidad gráfica
- Diversos tipos de caracteres
- Velocidad de transmisión entre 110 y 9600 baudios

Precio aproximado:
Con interface paralelo 134.900 pesetas
Con interface serie 149.900 pesetas
Nota: Al igual que las impresoras 8510, las impresoras 1550 SP y SR son versiones mejoradas.
- Precio:
  Con interface paralelo 149.900 pesetas
  Con interface serie 164.900 pesetas



Marca: C. Itoh
Modelo: 8600 P/R

Columnas: 80-136
Velocidad: 180 cps
Protocolo: Centronics o RS-232C
Buffer: 2 Kb
Arrastre: Fricción
Otras características:
- Cabezal de 18 agujas
- Bidireccional/ unidireccional
- Alta calidad a 90 cps
- Tipo "NLQ" a 45 cps
- Manejo del papel: Adelante, atras, micropasos
- Diversos tipos y tamaños de caracteres

Accesorios:
- Cargador automático de hojas
- Tractor de arrastre opcional

Precio aproximado:
Con interface paralelo 192.000 pesetas
Con interface serie 199.000 pesetas



Marca: C. Itoh
Modelo: 1570 SP/SR

Columnas: 136-233
Velocidad: 180 cps
Protocolo: Centronics o RS-232C
Buffer: 24 Kb
Arrastre: Fricción/Tracción (push/pull)
Papel: De 4 a 15,5 pulgadas
Otras características:
- Cabezal de 24 agujas
- Emula las características de la impresora tipo Margarita F-10 a 50 cps
- Gráficos de alta resolución
- Letra de calidad mediante el cambio de un interruptor

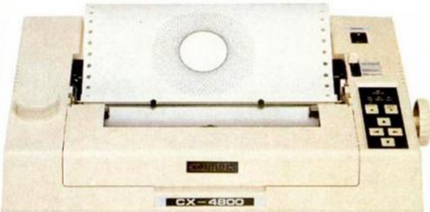
Precio aproximado:
Con interface paralelo 235.000 pesetas
Con interface serie 245.000 pesetas

Guía de impresoras PLOTTERS



Marca: Sony
Modelo: PRN-C41

Tipo: Plano
Velocidad como impresora: Media de 6 cps
Velocidad de la pluma: 57 mm./seg.
Area de dibujo:
Eje X 192 mm.
Eje Y 276,8 mm.
Incrementos mínimos (paso): 0,2 mm.
Precisión de posicionado: Menos de ±1%
Precisión de Imp. repetida: Menos de 0,2 mm.
Número de colores: 4
Tamaño del papel: Máx. hojas DIN A4
Protocolo: Centronics
Precio aproximado: 52.650 pesetas
Nota: Se suministra con adaptador de CA



Marca: C. Itoh
Modelo: CX 4800

Tipo: Tambor
Velocidad como impresora: 8 cps
Velocidad de la pluma: Máx. 120 mm./seg.
Area de dibujo:
Eje X 191 mm.
Eje Y 203,2 mm.
Incrementos mínimos: 0,05 mm.
Precisión de posicionado: 1% (100 mm. de recorrido)
Número de colores: 4
Tamaño del papel: 9,5 pulgadas para tractor y 9 para hojas
Protocolo: Centronics y RS-232C
Precio aproximado: 154.000 pesetas



Marca: C. Itoh
Modelo: CX 6000

Tipo: Plano
Velocidad como impresora: 3 cps
Velocidad de la pluma: Máx. 100 mm./seg.
Area de dibujo:
Eje X 288 mm.
Eje Y 216 mm.
Incrementos mínimos: 0,05 mm.
Precisión de posicionado: 1% (100 mm. de recorrido)
Número de colores: 6
Tamaño del papel: Máx. hojas DIN A4
Protocolo: Centronics y RS-232C
Precio aproximado: 198.000 pesetas


BIENVENIDOS A LA INGLATERRA FEUDAL
The EVIL Crown
Venciendo en los torneos reales podrás llegar a ser Rey.
¿Podrás apoderarte del resto del mundo medieval y adueñarte de la corona malvada?
APARICION SIMULTANEA EN INGLATERRA Y EN ESPAÑA
SPECTRUM
COMMODORE
MIND GAMES
LA CORONA MALVADA: UN JUEGO DE SIMULACION ANIMADA. LAS FUERZAS POLITICAS EN LA EDAD MEDIA. ¡FACIL DE JUGAR, MUY DIFICIL DE GANAR!
PREMIUM
SOFTWARE

WER

VARE, S.A.

- Tels. 232 24 61 - 232 25 52
RCELONA (SPAIN)

PREMIUM

SOFTWARE

CENTURY

SOFTWARE

¡¡¡ATENCION!!!

POWER, SOFTWARE, S.A.
Perseguirá por todos los medios legales a su alcance, cualquier tipo de piratería, comercialización, importaciones o duplicados ilegales que sobre sus programas se practique.



STRONTIUM DOG

Johny Alpha es un agente investigador destructor, un cazador de recompensas del futuro. Johny lleva consigo electrobengalas, que deslumbrarán a sus enemigos, obligándoles a dejar de disparar, y bombas de tiempo. Buena suerte en tu misión.



ASTRO BLASTER

5 Olas de ataque, 15 niveles de dificultad, fuego rápido, tormenta de meteoritos, alienígenas mutantes, joystick, bombas asesinas, gráficos a todo color en alta resolución, sonido, naves volantes, bolas de plasma, rayos laser.



PURPLE TURTLES

Basado en las legendarias tortugas púrpuras. Un juego para los jóvenes y demás personas que se hayan cansado de las palizas de los marcianitos.



BROAD STREET

... 7 Personajes, 10 acordes *perdidos, 15 horas,* 48 guardias de tráfico, 95 estaciones de metro, 45.000 m² de Londres, 7 millones de londinenses, 943 pantallas con trepidantes acciones.

Recorte y envíe este cupón a POWER, C/. Nápoles, 98, 1.° - 08013 BARCELONA, o llame al teléfono (93) 232 24 61

Deseo recibir los juegos que a continuación especifico, comprometiéndome al pago del importe de los mismos.

Nombre ______________________

Dirección ______________________

Teléfono ______________________

Firma:

☐ Contrarreembolso ☐ Adjunto Talón ☐ Giro Postal

Deseo recibir información de sus programas en: MSX ☐ AMSTRAD ☐

TITULO	SISTEMA	PRECIO UNIT.	CANTIDAD
LASER ZONE	Spectrum	975 Pts.	
GRIDRUNNER	Spectrum	975 Pts.	
FRENZY	Spectrum	1.275 Pts.	
ASTRO BLASTER	Spectrum	1.275 Pts.	
QUINTIC WARRIOR	Commodore	1.275 Pts.	
PURPLE TURTLES	Commodore	1.275 Pts.	
STRONTIUM DOG	Spectrum	1.800 Pts.	
STRONTIUM DOG	Commodore	2.100 Pts.	
FRIDGE FRENZY	Spectrum	1.800 Pts.	
BROAD STREET	Spectrum	2.400 Pts.	
BROAD STREET	Commodore	2.600 Pts.	
EVIL CROWN	Spectrum	2.400 Pts.	
EVIL CROWN	Commodore	2.600 Pts.	

Las Microestrellas

Gabriel Nieto

Hay quien asegura que en un lugar maravilloso, localizado en el interior de la imaginación de todos los micromaníacos, existe una ciudad que se llama MICROHOLLYWOOD, donde viven todas las estrellas que han protagonizado nuestros juegos favoritos.

Algunos son más conocidos que otros. Los hay que han intervenido solamente en un juego, otros, sin embargo, son asiduos en todas las producciones de alguna conocida compañía, pero todos en general tienen una personalidad propia, son ya parte de nosotros mismos, componen la gran familia de las microestrellas.



Un minero famoso

Al principio todos los juegos trataban de lo mismo, el tema espacial era el preferido de todos los programadores sin excepción. Los protagonistas eran naves con bellos diseños, dotadas de sofisticados armamentos, veloces como el rayo y con gran capacidad de maniobrabilidad, pero frías como el elemento mecánico que al fin y al cabo eran. Pero las tendencias cambiaron, los usuarios empezaban a estar hartos de matar tantos marcianitos y de volar entre explosiones continuas de fuego enemigo. Había que poner los pies sobre el suelo. Alguien, llamado Matthew Smith, pensó muy agudamente que mejor que en el suelo era debajo de éste. Y así se empezó a trabajar en la historia de un minero en apuros. El escenario elegido fue la mina de «surbitón», un lugar lleno de peligros con 20 pantallas distintas. El tema musical estaba listo, los malos de la historia también. Sólo faltaba el personaje central. Este debía de tener una personalidad propia que le diferenciara del resto. Matthew encontró su protagonista, se llamó Willy. Con su gorra, su pronunciado estómago y unos andares la mar de simpáticos, nuestro protagonista es el prototipo perfecto del hombre de la calle.

El juego fue un éxito sin precedentes y Willy se convirtió de la noche a la mañana en la primera Microestrella de nuestra historia.

Pero la aventura de Willy no terminaba allí. Muy pronto llegaría la segunda parte, «JET SET WILLY», que contra todo pronóstico, superaba al éxito anterior. El programa era producido por Software Projects, la compañía que a partir de ese momento se encargaría de dirigir el destino del minero-artista.

En este juego representa el papel de un nuevo rico que tras haber hecho fortuna en las minas de surbitón, lleva una vida de orgía y desenfreno, algo a lo que va a poner remedio su ama de llaves.

Es la culminación de la ascendente carrera de nuestro minero bonachón. Pero como a todas las estrellas también le llegó el ocaso. JET SET WILLY II representa el hastío, la repetición por norma y lo que es peor aún, una imaginación agotada. No sabemos cuanto tiempo seguirá en la brecha, pero sí intuimos que el final del pobre Willy, que tan buenos momentos nos hizo pasar, está cerca.

La gran avalancha

Manic Miner había abierto el camino que muy pronto seguirían otras compañías, de este modo hicieron su aparición en la escena del software nuevos personajes.

Uno de ellos es incluso anterior a Willy, nos estamos refiriendo a Horacio (Horace) que fue durante algún tiempo la estrella de la casa Psión y uno de los primeros en las producciones Sinclair. Pero Horacio no fue una gran estrella sino más bien una especie de pionero comparable a aquellos primeros artistas del cine mudo.

Otros personajes afloraron rápidamente motivados por el deseo de todos los productores de software de conseguir atraerse al usuario. Algunos, como Huchbank, procedían del mundo de la literatura. De esta forma se dio vida al personaje jorobado de la obra de Víctor Hugo, Notre Dame de Paris (Nuestra Señora de París). Tiempo más tarde se haría la segunda parte, pero la verdad es que tanto una como otra pasaron sin pena ni gloria.

Kosmic Kanga, el Oso Bobo, el mago de Sorcery, Borzak y el agente secreto de Ita Expres, Pogo, Gillians y algún otro por ahí desperdigado, son fruto de una avalancha de productos mediocres de los que sólo es posible salvar a BC



Bill, el curioso troglodita con un papel destacado en una de las últimas producciones de Imagine antes de que pasara a las filas de Ocean.

Los magos

Los magos también se hacen populares y representan su papel a la perfección. Las historias de magia y brujería ocupan un lugar importante entre los primeros juegos para Spectrum y es una vez más Imagine la que abre la brecha con Alchemist. El personaje es un alquimista que tiene la facultad de convertirse en Aguila y que va a enfrentarse al poderoso Warloc (un malo que aparecerá también en otros juegos) en una excitante aventura con un escenario de cavernas laberínticas.

El Alquimista no volvería a aparecer en ningún otro juego, pero otra casa, Hewson Consultants, aprovechando el éxito de éste, crea su propia historia y su propio personaje. Estamos hablando de Maroc, el protagonista de Avalon. Una vez más repite el éxito y se decide probar suerte con una segunda parte que en esta ocasión se llama Dragontorc y que es prácticamente igual que el anterior.

Hace poco surgía la Bruja de Cauldrón, uno de esos personajes que parece sacado de los cuentos infantiles y que viene a ser continuación de un género que está muy lejos de desaparecer.

Orestes y Ocean

A Ocean nunca se le ha dado bien eso de crear personajes, véase el caso de Huchbank, Gillians (un claro intento de crear una especie de Manic Miner), el héroe de Kong (que tiene cierta similitud con Gillians), el atleta del Decathlon, Pud, y mucho más recientemente Roland, este último el más destacable de todos.

A pesar de ello hay un personaje creado por Ocean que es una auténtica maravilla. Se trata de Orestes el protagonista de Gift from the Gods.

El juego, basado en una apasionante historia mitológica, nos muestra a un Orestes pletórico de fuerza con rasgos muy bien definidos y una puesta en escena (se entiende escena de software) verdaderamente brillante.

Sin embargo, Ghift from the Good no tuvo el éxito esperado, quizá por su enorme complejidad.

Las estrellas solitarias

Como en el mundo del Cine, aquí también hay actores (microactores en este caso) que sólo han intervenido en un juego y que, nacidos para un solo producto, también han dejado su huella más o menos profunda.

Entre estos los ha habido procedentes del mundo animal. Es el caso del ratón Jasper, y del gusano Wringler, este último con una mascota de promoción incluida.

Otros han tratado de ser los sucesores del minero Willy con más o menos fortuna. De todos ellos el más representativo sea quizá Ted el Técnico, un personaje clarísimamente inspirado en el anterior que nos hacia revivir otros tiempos llenos de nostalgia en un juego bastante divertido que, sin embargo, tampoco es demasiado conocido en el interior de nuestras fronteras.

Kokotoni Wilf es uno de esos héroes solitarios de la escuela también de Manic Miner, pero que nos traslada a un mundo fantástico en una historia a través del tiempo. El personaje es un angel

en busca de siete estrellas que están repartidas por siete épocas distintas.

Una de las figuras del software más peculiares que surgieron en esta época fue sin lugar a dudas Trahsman, que encarnaba el papel de un basurero en el complicada tarea de cada día. La idea fue de lo más original y Trashsman tuvo una estupenda acogida. Precisamente por eso los señores de New Generation pensaron que sería conveniente hacer la consabida segunda parte y así trasladaron al basurero (que por cierto ahora era rico) a diferentes países del mundo en los que seguiría practicando la noble profesión de recogerlo todo.



Las aventuras de Monty ocupan también un lugar importante aunque no muy destacado. El personaje procede también del reino animal y protagoniza una serie de aventuras que sin llegar a ser éxitos notables, si son el cambio de indudable calidad, Monty is Inocent y Monty Mole son dos producciones de Gremlins Graphics en las que se luce nuestro héroe. Casi paralelamente a éste aparece un competidor de la misma compañía, Sam que, la verdad, no tiene demasiada resonancia.

Los personajes del cine en el mundo del software

El cine, que siempre ha presumido de beber de otras fuentes artísticas y literarias, también ha encontrado en el software un duro competidor que se inspira a su vez en éste.

Desde Hulk hasta el actual Rambo han ido desfilando por la pantalla, que genera nuestro ordenador, personajes populares del mundo del cine, el comic y la literatura: Sherlock Holmes, Bruce Lee, los muchachos de Ghostbusters, Hulk, Spiderman, James Bond, Rocky, Robind Hood y Superman.

También el teatro, la televisión y la música han tenido un lugar importante con las versiones del Rocky Horror Show, la serie Minder y la película de Paul McCartney.

Actualmente, la compañía US GOLD trabaja en un ambicioso proyecto para adaptar a juegos de ordenadores las películas y programas más populares de Walt Disney: EL LIBRO DE LA SELVA, EL REGRESO DEL MAGO DE HOZ, EL CALDERO NEGRO, MICKEY MOUSE, EL PATO DONALD y PLUTO.

Sabreman, la estrella de ultimate

Después de Willy y a pesar de la invasión de personajes, la verdad es que no vuelve a haber una estrella hasta Sabreman, un producto de la casa Ultimate que a la postre habría de convertirse en la empresa más importante de Software para Spectrum del mundo.

Sabreman es la clave, el principio de una nueva generación de software, representa la aventura por la aventura, un nuevo concepto de programación y es, además, el primer intento serio de crear una serie en forma de trilogía que acabará siendo archipopular a todos los niveles. El escenario es en esta ocasión la selva, y Sabreman representa a un explorador de los de «salacof» incluido que tendrá que enfrentarse a los innumerables peligros que le aguardan en la jungla, hasta llegar a la guarida de Sabre Wulf donde hallará la primera de las tres claves que le conduzcan hacia el final de la historia.

Sabre Wulf es un éxito sin precedentes y se convierte rápidamente en el número uno sin discusión. Y de forma paralela el nombre de Ultimate empieza a sonar con fuerza en el mundo del Software.

Muy pronto llegaría la segunda parte, Underwurlde, que contra todo pronóstico (por eso de que segundas partes nunca fueron buenas) es también un notable éxito.

Y casi seguido aparece la tercera parte que es la culminación de la obra Ultimate, Knight Lore, un programa que utiliza por primera vez la técnica filmation y que es, además, la última parte de la trilogía.

Sabreman es un ídolo, una revolución técnica, la culminación de un estilo, pero Ultimate no sabe asimilarlo y se olvida de su estrella (su buena estrella) con un nuevo lanzamiento, Alien 8 que es la gran decepción. Y no porque el programa sea malo, porque la verdad es que está muy bien hecho, sino porque Ultimate trata de repetir en el pequeño Robot el éxito del inimitable Sabreman, trasladando la misma historia y casi el mismo decorado a una galaxia desconocida que parece haberse convertido en el único plató donde trabaja la compañía. Y prueba de ello es Night Shade, un programa que ha encendido la mecha y que de seguir así acabará minando la estructura de una compañía de software que llegó a ser la mejor del mundo.



Wally

Poco tiempo después de que Sabreman se convirtiera en el ídolo de los spectrunmaniacos, irrumpía en la escena softwarera un tipo con cara de currante (de los de toda la vida), barriga pronunciada, gorrilla y andares pausados. Su nombre es Wally y pronto se hará muy popular.

El primer juego en el que interviene es «Automania» donde Wally es un mecánico que tiene que luchar con los elementos que le rodean y dificultan su misión, que no es otra que reparar diez coches. El programa está en la línea de Manic Miner pero con una concepción distinta en lo que se reifere al tratamiento de las pantallas.

Automania, a pesar de su indudable calidad, no es un juego de los que marcan una época, pero al menos si es el inicio de un estilo que acabará siendo el más importante de la escena del software.

La continuación es «Pyjamarama» y aquí sí, la empresa Micro Gen da el golpe de gracia con un programa tremendamente original que obtiene un clamoroso éxito. El protagonista es también Wally y se convierte rápidamente en un personaje famoso.

El juego a pesar de su buena acogida no es demasiado conocido en España y tarda algún tiempo en conquistarnos. Quizás influyera la portada del juego que no era tan llamativa como otras, a las que estábamos más acostombrados por aquella época.

Y llega la tercera parte, «Everyone's a Wally», el mejor juego de la serie y la

obra cumbre de Micro Gen. En este juego descubrimos además nuevos personajes que se interelacionan entre sí pero de una forma casi real. El programa podía definirse como una película de software en la que todo ocurre con un sentido eminentemente lógico.

Wilma, Tom, Dick, Harry y el pequeño Herbert componen el reparto principal del juego. Cada uno con una personalidad propia que les define, pero de todos ellos va a ser Herbert, que es el que menos interviene, el que acabe saltando a la fama, y se convierta en el protagonista de la nueva producción de la compañía, Herbert's Dummy Run, en el que también intervienen Wally y Wilma, aunque en esta ocasión en un papel secundario.



Micro Gen es quizás de todas las compañías de software la que ha actuado con más inteligencia creando una plantilla de personajes que utilizan de una forma flexible para evitar de este modo caer en la repetición.

Actualmente, está preparando la segunda parte de «Everyone's a Wally», que llevará el nombre de «Three Week in Paradise» y en la que veremos a todos los personajes juntos otra vez.

El secreto de Micro Gen ha estado, hasta ahora, en dar a Wally en cada uno de los juegos en los que ha aparecido un papel distinto conservando, sin embargo, la misma personalidad.

Johny Jones el héroe de Dinamic

Johny Jones es la versión hispana del héroe aventurero de los juegos ingleses, que nace con la compañía Dinamic y que va a ser el responsable del encumbramiento de ésta.

Su primera aparición es en Saimazoom, donde tiene que encontrar los cuatro sacos de café que se encuentran perdidos en la jungla. El juego está inspirado en Sabre Wulf y sale algún tiempo después de éste.

Más tarde llegaría Babaliba, una aventura inspirado en un palacio moro, de los de las mil y una noche, en la que nuestro amigo Jones tiene que raptar una princesa, rescatar a un amigo y encontrar un tesoro.

Tanto un juego como otro tienen un corte parecido, aunque en el segundo se superan algunas cosas que en el primero habían sido fruto de la inexperiencia y se consigue crear un producto de bastante calidad, convirtiendo a Johny Jones en una estrella en alza.

Pero la revelación es «Abu Simbel Profanation», un juego de bastante calidad que se convierte rápidamente en un número uno sin discusión. La historia está ambientada en Egipto y para su realización los programadores se basan en una amplia bibliografía, como por ejemplo el libro Templos y Sabios.

El programa supone la culminación de la carrera de Johny Jones en España y el lanzamiento internacional del personaje, ya que atraviesa nuestras fronteras y llega hasta Inglaterra de la mano de Gremlin Graphics que le prepara una campaña por todo lo alto.

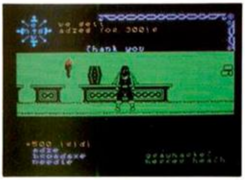
Dinamic logra así situar a su estrella en el olimpo del software. Lo que habrá que ver es por tanto cuánto tiempo es capaz de mantener a su personaje sin caer, como han hecho otros, en el hastío.



Cuchulain

Cuchulain es una mezcla entre ficción y leyenda, y Gargoyle Games lo sabe, por eso nos presenta a su personaje en un ambiente muy propicio para la aventura, entre pergaminos y mapas misteriosos que componen la ambientación de una historia basada en una precisa leyenda llena de emoción.

El programa es Tir Na Nog y tras él llegará Dun Darach una historia más bonita aún que la primera en la que se ha aumentado el reparto, los medios y la complejidad del argumento para crear un juego en el que reina la fantasía. Es una especie de historia mitológica que llega profundamente a todos los amantes de las historias fantásticas.



Los nuevos valores

Hasta aquí llega la historia de unos personajes que son ya parte de nosotros mismos, que nacieron para entretenernos y se han convertido en algo más que en frutos de la imaginación de algún buen programador. El usuario ya no espera el nuevo programa de Ocean, de Ultimate, de Micro Gen, ni siquiera le preocupa que el programa que tiene en sus manos sea de Matthew Smith o de Víctor Ruiz, porque lo que espera en realidad encontrar al acabar de cargar una cinta es a su viejo amigo Sabreman o el escenario donde vivirá una nueva aventura junto a Cuchalain.

Desde el primer juego de Spectrum hasta hoy, ha llovido bastante y ahora nuevos nombres comienzan a despuntar: Dan, de Dynamite Dan; Charlie, el astronauta de Nodes of Yesod; Bob, de Bounty Bob... pero de todos los personajes que hemos conocido desde entonces, compañeros de aventura como Willy, Sabreman, Johny Jones, Wally o Cuchulain, ha habido realmente muy pocos.

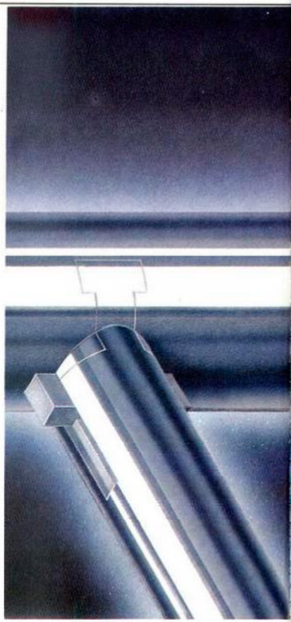
Ellos viven en el Olimpo del Software.



GENS y MONS-3

J. M. LAZO y Heliodoro M. SALINAS

Cualquier comienzo vale para definir este maravilloso programa, sin dudarlo, el mejor que existe para microordenadores, pero el lector ya se estará preguntando de qué estamos hablando. Pues bien, estamos hablando de un ensamblador que posee una serie de facilidades que hacen que el aprendizaje del código máquina sea tan fácil como el del Basic.



¿Y qué es eso de un ensamblador? Un ensamblador es un «toolkit», es decir, un programa herramienta, que se utiliza para poder programar en el árido lenguaje máquina de una forma más sencilla, utilizando en vez de números a palo seco, una serie de palabras (nemónicos o nemotécnicos) las cuales el ensamblador se encarga de pasar a números. La lista de estas palabras se llama Código fuente, y es el programa en CM que nosotros podemos escribir, y la lista de números resultantes después de ensamblado el Código fuente se llama Código objeto.

Para entrar en materia

Nuestro propósito al escribir este artículo es explicar, de la forma más amena y didáctica, la utilización del GENS.

En un principio hay que decir que el programa es totalmente reubicable, esto quiere decir que funciona en cualquier zona de la memoria que se halle libre en el momento de cargarlo.

Para cargarlo en el Spectrum teclearemos LOAD " " CODE dirección, donde la dirección es de la misma forma la zona de memoria donde ubicaremos el programa.

Una vez cargado el programa nos pedirá Buffer size? de momento pulsa ENTER directamente, y entramos en el editor, es decir, el ordenador esperará nuestras órdenes.

Vamos a picar el siguiente programa en el ordenador, y de paso aprenderás como funciona el GENS, casi sin darte cuenta, sigue nuestras explicaciones y verás qué pronto aprendes.

Primero tecleamos directamente: I 1Ø, 1Ø. Esto significa lo siguiente: el comando "I" es uno de los muchos que tiene el GEN y sirve para generar los números de línea automáticamente, a continuación del comando se ponen dos números, el primero es el que indica a partir de que número se empezará la secuencia, y el segundo indica el incremento. Se puede eludir uno de ellos (o los dos) asignando el programa automáticamente, el último que hayamos tecleado.

Cuando esté hecho veremos que aparece un número (1Ø) y el cursor a la derecha, en este momento estamos dispuestos para empezar a introducir el programa.

El programa

Introdúcelo pues, procurando no equivocarte nada en su contexto.

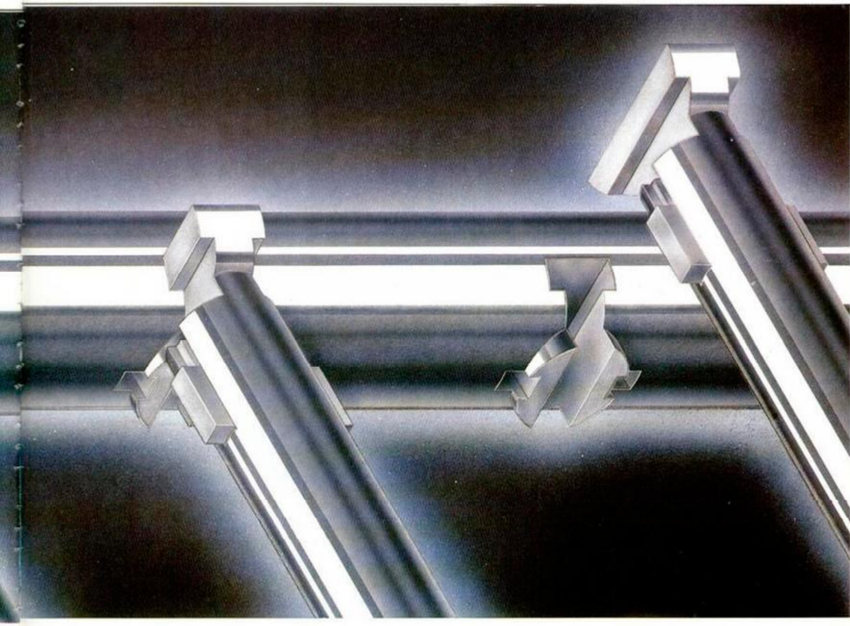
```
1Ø ORG 5ØØØ
2Ø ENT $
3Ø START LD HL,:4ØØØ; CARGAR EN HL EL PRINCIPIO DEL BUCLE
4Ø LOOP LD A,R; CARGAR A CON UN NUMERO ALEATORIO
5Ø LD (HL),A; CARGAR EL NUMERO ALEATORIO
6Ø INC HL INCREMENTAR EL PUNTERO
7Ø LD A,:58
8Ø CP H
9Ø JR NZ,LOOP; SI NO HA LLEGADO AL FINAL VOLVER
1ØØ LOOP1 LD A,191
11Ø IN A,(:FE); LEER SEMIFILA "H-ENTER"
12Ø BIT 2,A
13Ø JR Z,LOOP1; SI ESTA PULSADA "K" PARAR
14Ø BIT 1,A
15Ø JR NZ,STAR; SI NO ESTA PULSADA "L" VOLVER A STAR
16Ø REY ; RETORNAR
```

Ensamblado

Pulsa a continuación la orden "A". Este comando sirve para decirle al ensamblador que puede empezar a ensamblar (traducir) el código fuente, después de haberlo hecho ya dispondremos del código objeto, el cual sirve para correr el programa.

Cuando hayas dado la orden el programa te preguntará Table size:? Pulsa de momento ENTER directamente y entonces te preguntará Options:? A esto responde con "1" y pulsa ENTER.

Entonces el ensamblador nos dirá que hay un error en la línea 6Ø. Si queremos continuar el ensamblado a pesar del error, pulsamos ENTER, y si queremos corregir el error damos a la "E". De



momento vamos a estudiar el error para tratar de corregirlo.

Vemos que es el error 6, si miramos el manual veremos que el error 6 es Operando ilegal. Esto a veces en vez de ayudar confunde más, pero si examinamos la línea atentamente veremos que antes del comentario falta un punto y coma. Este separador sirve para decirle al ensamblador que todo lo que viene a continuación es un comentario, algo así como las sentencias REM del Basic. Como falta, el ensamblador no entiende lo que viene a continuación.

Para corregirlo damos a la «E» dos veces. Este comando sirve para editar una línea del programa y poderla corregir. Precisa un argumento numérico detrás del comando, que es la línea que precisamos editar, por lo cual tecleamos E6Ø y pulsamos ENTER.

Inmediatamente saldrá una copia de la línea 6Ø y debajo el cursor. Dentro del comando «E» hay otros varios subcomandos:

El espacio sirve para avanzar el cursor sin modificar nada, démosle pues, hasta que llegue a la posición entre HL e INCREMENTAR. En este sitio, es preciso «I»nsertar un punto y coma para aislar el comentario del operando, por lo que pulsemos la «I», vemos de esta manera cómo los programadores que hicieron este programa, «GENS», imprimieron en él una facilidad de uso tremenda, dando a cada comando la inicial de lo que significan.

Una vez pulsemos «I», veremos cómo el cursor se convierte en un asterisco indicando con esto que estamos en el modo de inserción, tecleamos sin más el punto y coma, y damos a ENTER dos veces. Esto es necesario ya que la primera es para salir del modo inserción, y la segunda del modo edición.

Vamos a ver si la línea 6Ø en nuestro programa se ha puesto bien, para ello «L»istamos el programa con la orden L y vemos como la línea 6Ø ya está bien.

Ensamblamos otra vez con el comando arriba explicado y respondiendo con ENTER a la primera pregunta y con 1 a la segunda, nos encontramos con otro error, esta vez en la línea 16Ø.

Ahora es el error 3, sentencia mal construida. Y otra vez el error poco aclara nuestras dudas, si miramos la línea veremos que el RET tiene la T cambiada por una Y, pero esta vez vamos a corregirlo de otra forma para aprender otro comando.

Da una vez a «E» con lo que tendremos el cursor en la pantalla y tecleamos «F1,1ØØØ,REY,RET». Cuando lo hayamos hecho estaremos en el modo de Edición con la línea 16Ø y el cursor en medio.

El comando «F» sirve para buscar («F»ind en inglés), etiquetas y otras literales dentro del programa. Lo que hemos hecho en esta ocasión es buscar desde la línea 1 a la línea 1ØØØ la literal «REY» y como último parámetro del comando «F» hemos tecleado RET.

En el modo de Edición, si damos al comando «S»ustituir, sustituimos la primera literal del comando «F» por la segunda, en esta caso «REY» por «RET», de esta forma hemos sabido una función muy interesante que nos permitirá buscar dentro de un programa muy grande una etiqueta en particular.

Volvemos a ensamblar, de la forma explicada arriba, y al ensamblador no

se le ocurre otra cosa que decirnos «Bad ORG!», este error (por lo menos más expresivo que: error n...), significa que en el seudonemónico ORG el argumento que le acompaña no está bien colocado. El ensamblador a la hora de ensamblar calcula que el código objeto no vaya colocado encima de él mismo o del código fuente, en caso de que sea así detiene el ensamblado y vuelve al editor.

Si no te ha dado el error significa que el ORG está bien colocado.

Si el ensamblado ha transcurrido sin errores saldrá un listado del código fuente y un listado de las etiquetas por orden alfabético. Encima de las etiquetas se hallan las que no estén definidas (si es que hay alguna). En nuestro caso sí, y podemos leer «*WARNING* STAR absent», esto significa tanto como: CUIDADO STAR ausente. Este



error se produce cuando intentamos utilizar una etiqueta que no hayamos definido dentro del programa. Veamos a nuestro programa para saber dónde puede estar el error. Lo listamos (con «L») y vemos en la línea 3Ø la definición de la etiqueta «START», y si seguimos mirando en la línea 15Ø, se haya una instrucción «JR NZ, STAR». Como podemos ver le falta la «T» del final, que se la podemos poner de cualquiera de las formas aprendidas anteriormente. Por ejemplo:

Edita la línea 15Ø (E15Ø), entonces nos vamos con el espacio hasta encima del punto y coma. Y damos al comando «I», pudiendo en este momento insertar la «T» que falta. Por último pulsamos dos veces ENTER.

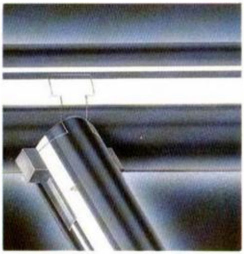
Si ensamblamos el programa ahora, veremos cómo no salen más errores, a no ser que al copiarlo hayamos cometido alguno.

Comandos del Cassette

Una vez ensamblado el programa sin errores es muy conveniente volcar el listado fuente en cinta para no perderlo en caso de que el ordenador se «cuelgue» al correrlo.

Podemos grabarlo de dos formas distintas, para tenerlo como programa, o como subrutina, teniendo la opción de poderlo cargar luego para juntarlo con otro programa.

Para grabarlo como programa procedemos con el comando «P» al cual tenemos que añadir dos parámetros numéricos, el primero es la primera línea que queremos grabar, y el segundo es la última, así que para grabar el nuestro ahora, podemos teclear «P1,1ØØØ,Nombre», donde «Nombre» es el nombre que queremos darle al programa. No te olvides de poner el cassette en modo REC antes de dar a ENTER, ya que esta opción no pide confirmación de pulsar una tecla. Por último, podemos grabarlo como subrutina, con el comando T, el cual no precisa ningún argumento des-



pués, sólo pulsar «T» y ENTER, entonces se volcará en cinta el programa, pero de una forma especial, por bloques, que luego podemos cargar con la orden de ensamblador «F».

Para cargar Códigos fuente en el ordenador, se usa el comando «G», teniendo éste (si se desea) un parámetro que es el nombre y que se pone después de dos comas y del comando, de la forma: G,,Nombre.

Table size? Options:? Buffer size?

Anteriormente pasamos respondiendo a unas opciones, y no explicamos su cometido, en este momento vamos a hacerlo.

Cuando entramos al programa (GENS) por primer vez nos preguntará: Buffer size? O lo que es lo mismo: ¿Tamaño del Buffer? a la que respondimos pulsando ENTER directamente. El Buffer es el sitio donde el programa almacena los pequeños trozos en que graba el código fuente, cuando ha de hacerlo, pues bien, lo que nos pregunta al principio es el tamaño de estos trozos, teniendo en cuenta que cuanto más pequeños sean éstos más memoria ahorraremos con el comando de ensamblador «F», pero más espacio ocupará en cinta y por lo tanto, más tiempo tardará el ensamblado desde cassette.

Si a la pregunta respondemos con «Ø» el tamaño del buffer será de 64 octetos, si con «1» será de 256, si con «2» de 512, etc. Si respondemos con ENTER directamente se le asignará el número «4», o sea, 1Ø24 octetos.

Después a la hora de ensamblar, nos preguntó Table size?, o lo que es lo mismo: Tamaño de la tabla. En primer lugar, esta pregunta se refiere al sitio que va a dejar en memoria para la tabla de símbolos (etiquetas). Si respondemos con ENTER el programa le asignará un valor medio, que puede va-



ler, pero que a veces es muy pequeño (depende de la profusión con que hayamos usado las etiquetas). El número que introduzcamos será el que reservemos en memoria (en octetos) para las etiquetas.

Después nos preguntó Options?, o sea, ¿Opción? Esto tiene un poco de más tela, hay 63 opciones disponibles, depende de si queremos listado o no, de si vamos a producir código objeto o no, etc. Dimos a la opción «1» que produce: un listado de la tabla de etiquetas después del ensamblado, produce el código objeto después del ensamblado, lista el programa, no lo lista por la impresora, hace caso del pseudonemónico ORG, y por último, revisa la zona del código objeto por si se solapa con el propio GEN.

¿Qué cómo sabemos esto? pues muy fácil, cada bit del número que tecleemos como respuesta, indica una cosa:

— El primero: (1) si está alto da la tabla de etiquetas al final.

— El segundo: (2) si está alto no produce el código objeto.

— El tercero: (4) si está alto no lista el programa.

— El cuarto: (8) si está alto lista el

programa por impresora.

— El quinto: (16) si está alto no hace caso del nemónico ORG, poniendo el código objeto detrás de las etiquetas.

— El sexto (y último): (32) cuando está alto produce un ensamblado rápido, no verificando la zona donde se va depositando el código objeto.

Después de este rollo os aconsejamos una cosa:

La primera vez que ensambléis el programa dad a la opción 5 que no lista el programa, lo cual puede hacerse pesado, pero lista las etiquetas al final. Cuando hayas ensamblado más de una vez el mismo programa, dad a la opción 36, que es la que tarda menos tiempo, y si queréis sacarlo por la impresora dad a la opción 9.



Correr el programa

Después de todas estas explicaciones necesarias para comprender el correcto funcionamiento del programa (GENS), vamos a volver a la práctica, corriendo el anterior programa. Para ello si no lo has hecho aún, ensámblalo con el comando «A» respondiendo a la primera pregunta con ENTER y a la segunda con «5» y entonces podrás ver que en la pantalla pone «Executes:xxxxx», el número significa la dirección de comienzo del programa en máquina, y que coincide con lo que hayamos puesto en el pseudonemónico «ENT», en el ejemplo hemos puesto «$» esto es así porque deseamos entrar al programa ahí exactamente.

Para probar podemos ejecutarlo desde el Basic con la sentencia RANDOMIZE USR dirección, o similar, donde dirección es la dirección donde esté ubicado el código objeto, o podemos probarlo desde dentro del propio ensamblador con el comando «R» a palo seco. Entonces el programa funcionará volviendo al ensamblador en el momento en que se encuentre con la instrucción RET (en el programa de demostración pulsando la «L»).

Breve glosario de comandos

Por último, vamos a daros una pequeña lista de comandos y su utilización.

Para volver a ambiente Basic: el comando «B» sin ningún parámetro.

Para borrar líneas, el comando «D» con dos parámetros, el primero es la primera línea a borrar, y el segundo la última.

Para listar el programa, el comando «L» con dos parámetros, el primero es la primera línea a listar, y el segundo la última.

Para renumerar el programa el comando «N» con dos parámetros, el primero la primera línea que queremos que nos quede, y el segundo el paso entre líneas.



Para cambiar el separador de parámetros en los comandos: el comando «S», a continuación dos comas (o el separador actual) y a continuación el número que queremos.

Para listar en impresora el listado fuente el comando «W», con dos parámetros al igual que el «L»ist.

Para cambiar el número de líneas que son listadas de un golpe con el comando «L» o «W» el comando «K» seguido de el número de líneas.

Para ver dónde se ubica el código fuente el comando «X»

Creemos que esta pequeña iniciación servirá para que se decidan los que estén dudosos ante el tema, y de ayuda para los que ya han empezado.

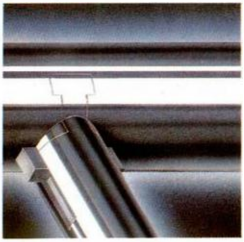
Mons-3

Presentamos en esta ocasión un programa que hará las delicias de muchos programadores en lenguaje máquina y de muchos otros que aún no están introducidos en este lenguaje de programación, y que a través de este artículo esperamos que vean el camino mucho más sencillo, ya que van a tener una buena herramienta de trabajo: el desensamblador MONS-3.

Este desensamblador es el complemento del ensamblador GENS-3, y juntos forman una pareja perfecta.

Pero... ¿qué es un desensamblador?

Como muchos de vosotros sabréis, un desemsamblador sirve para inspeccionar un programa en código máquina situado en la memoria del ordenador (ya sea ROM o RAM), y ver su listado no como una serie de números, sino con una serie de nemónicos un poco más comprensibles para el ser humano. Pero este desemsamblador también reune otras funciones, que ya reseñaremos después.

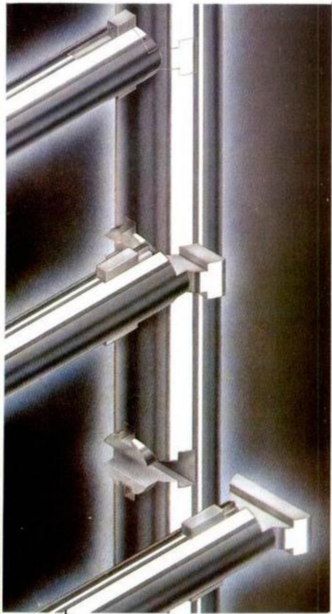


Empecemos

Lo primero que hay que hacer para usar cualquier programa es cargarlo. Y allá vamos.

La primera peculiaridad que se encuentra en este programa es que es totalmente reubicable. De otra forma: lo podremos cargar en cualquier parte de la memoria del ordenador (siempre que no estorbe al BASIC, variables del sistema, etc.). En nuestro ejemplo lo cargaremos en la posición 40000 con el comando: LOAD «MONS3» CODE 40000.

Cuando haya acabado de cargar, podremos acceder a él tecleando RANDOMIZE USR 40000 (conviene apuntar este número en algún sitio, aunque por otra parte es fácil de recordar). Pero debemos tener en cuenta que, cuando hayamos vuelto al BASIC desde el MONS-3 y queramos volver a entrar en el programa teclearemos RANDOMIZE USR dirección+29, donde la dirección es la dirección de carga del programa.



Cuando ya hayamos realizado la entrada en el programa, podremos ver la pantalla dividida en varios espacios:

— En la primera línea de la pantalla, a la izquierda, hay un número hexadecimal: este número indica la posición de memoria a la que está apuntando el MONS-3. A su derecha está el contenido de esta posición y un poco más a la derecha está su traducción a nemónico.

— Zona de registros: En la parte superior izquierda, debajo de la primera línea, se encuentran los nombres de todos los registros, y a su derecha, el valor que les corresponde en ese momento. Veremos cómo se pueden modificar los valores de éstos. A la derecha de esto se encuentra el valor de las posiciones de memoria a las que apuntan los registros, y las seis posiciones siguientes.

— Zona central: Aquí están una serie de números que representan posiciones de memoria y el valor que contiene cada una. Hay una de estas posiciones de memoria, señalada en el centro de la pantalla con ">   <", que, si os dáis cuenta, es la que está en la parte superior de la pantalla. Esta sección de la presentación sirve para ver todas las posiciones de memoria adyacentes a la que tenemos elegida.

— Zona de comandos: Está en la parte inferior de la pantalla, y es donde vamos a introducir todas las instrucciones necesarias para el funcionamiento del programa.

Comandos

Primero explicaremos los comandos de desensamblado, poniendo el puntero de memoria (pulsando M y después un número) en la memoria ROM, en la posición Ø4C2 (siempre en hexadecimal). Pulsamos symbol shift+4 y podremos ver el desensamblado de la memoria a partir de esta posición (comienzo de la subrutina SAVE del Spectrum). Si queremos ver más posiciones de memoria, pulsaremos cualquier tecla y seguirán saliendo más partes de la memoria desensambladas. Para finalizar, volveremos a pulsar symbol+4.

Podemos ver las posiciones de memoria presentadas en decimal pulsando symbol+3, y para tenerlas otra vez en hexadecimal pulsar symbol+3 de nuevo. Para volver al BASIC (cuando hayamos termindo), habrá que pulsar EDIT (Caps Shif+1).

Para traducir un número decimal a hexadecimal, pulsar H, introducir el número decimal y pulsar ENTER, con lo que aparecerá el número en hexadecimal.

Hasta aquí son las funciones que realiza un desensamblador, pero ahora empezaremos con las grandes facilidades de uso del MONS-3.

Pongamos el Memory Ponter (puntero de memoria) en la posición ØØ96 (esto se hace pulsando M y después la posición de memoria que queremos observar). Pulsamos la L y podremos ver un listado de la memoria en ASCII, a partir de donde están almacenados en ROM los tokens del BASIC. Para salir de esta opción pulsar Caps shift+5. Esta función del MONS-3 se puede usar para buscar en un programa una parte literal (o sea, un texto).

Otro comando muy útil es el comando G. Nos permite buscar en la memoria del ordenador una secuencia de números. Se utiliza de la siguiente manera: Tenemos un programa el cual es llamado desde otra parte de la memoria. Nuestro programa está situado en la posición C35Ø, y queremos saber desde dónde es llamado. Entonces teclearíamos G (a continuación nos salen dos puntos) y después el byte menos significativo de la dirección (5Ø), pulsamos ENTER, introducimos el byte más significativo (C3) y pulsamos dos veces a ENTER. Inmediatamente el programa buscará estos dos números juntos en memoria y nos colocará el memory pointer en donde ha encontrado los dos números. Sólo tendríamos que desensamblar y ver qué instrucciones hay en esas posiciones de memoria.

Si queremos seguir buscando la misma secuencia de números, pulsaríamos N y el ordenador seguiría buscando hasta encontrar otra secuencia de números igual a la que ha buscado anteriormente.

Registros

Más arriba hablamos sobre los registros, que están todos escritos en la parte superior izquierda.

Para variar el valor de un registro hay que hacer lo siguiente:

Por lo pronto, habrá que apuntar al registro con el indicador ">". Esto se hace pulsando el punto (symbol+m), y podremos ver cómo el indicador se va moviendo de unos a otros registros. También podremos apreciar que el indicador nunca apunta a SP ni a IR. Esto se produce para que no podamos variar el puntero de pila (SP) en cualquier instante, porque podríamos provocar un crash del sistema. Tampoco podemos modificar IR porque entonces modificaríamos el vector de interrupción I, y podríamos obtener las mismas consecuencias que al modificar SP.

Después de haber apuntado con el indicador a los registros que queremos modificar, lo único que hay que hacer es introducir el nuevo valor de los registros y, en vez de pulsar ENTER, pulsaremos el punto, con lo que el nuevo valor queda presentado en pantalla.

Un ejemplo

Proponemos un pequeño programa en código máquina muy sencillo (que se pude teclear con el GENS3), pero que servirá muy bien para explicar algunas de las más interesantes funciones que realiza el MONS3.

Lo único que hace este programa es cambiar el color del borde entre blanco y negro, pasando por toda la gama de colores:

```
1Ø          ORG 5ØØØØ
2Ø          LD B,8
3Ø  LOOP    LD A,B
4Ø          DEC A
5Ø          OUT (1FE),A
6Ø          DJNZ LOOP
7Ø          RET
```

Después de haber cargado el GENS3 en la posición 3ØØØØ y el MONS3 en

la posición 40000 (es conveniente cargar los dos programas siempre a la vez) tecleamos el programa anterior y lo ensamblamos. Nos vamos al MONS3, ponemos el memory ponter en la posición 50000 y podremos ver el programa en código máquina que hemos generado. Con él podremos practicar con las explicaciones que se darán a continuación.

Breakpoints

Breakpoint, o punto de ruptura, es el punto en el cual un programa que estemos corriendo en el ordenador se detiene para volver al desensamblador, e informarnos del valor de los registros en ese momento.

Para colocar un breakpoint en nuestro programa hay que proceder de la siguiente manera:

Observaremos el sitio donde queremos que se pare la ejecución del programa, por ejemplo en la instrucción OUT (#FE), A. Entonces pondremos el memory pointer justo después de esta instrucción (en la instrucción DJNZ LOOP). Pulsamos la W y vemos como la instrucción que estaba en ese momento es cambiada. Esto no importa porque luego la recuperaremos. Entonces pulsamos la J, que sirve para saltar a un programa desde el MONS, introduciendo luego la dirección a donde deseamos saltar, en este caso C350 (insistimos en la necesidad de la entrada de números en hexadecimal) y veremos como el color del borde cambia a blanco y el programa se detiene volviendo al desensamblador. Aquí podemos observar los valores de todos los registros, y también del registro F, que se nos presentará como varias letras a la derecha de el (cada letra corresponde a un flag o banderín del registro de estados del microprocesador. Si el banderín está alto, la letra es presentada en pantalla, y si está bajo ésta será omitida). Esta opción es muy útil ya que podemos ver la ejecución de una parte del programa y también observar sus resultados.

Pero hay todavía otra opción más interesante si cabe...

Paso a Paso

En muchos libros de texto no referidos únicamente a un ordenador se pueden ver programas-ejemplo que se corren paso a paso y en cada uno de ellos se observa su resultado en los registros y banderines. Pues bien, el MONS3 puede realizar esta opción, tanto en RAM como en ROM.

Vamos a experimentar en nuestro pequeño programa. Para ello colocamos el memory pointer en la posición 50000 (C350) y también debemos poner en la misma posición el program counter (registro PC) y después pulsar symbol shift+z. Podremos ver como el PC ha aumentado en una instrucción, y los registros se han modificado de acuerdo a la instrucción, y los registros se han modificado de acuerdo a la instrucción realizada. Si seguimos pulsando continuamente la Z, se ejecutará una por una todas las instrucciones del programa, y observando que, cada vez que pasemos por el bucle, el color del borde variará.

También se puede correr paso a paso cualquier subrutina de la ROM (por ejemplo la de SAVE, situada en la posición de memoria 4C2) y ver los efectos que acarrea en los registros, al mismo tiempo que podemos modificar los registros y ver en qué influyen sobre el transcurso del programa.

Comandos de cassette

Otra opción muy poderosa, y con esto acabamos, es la de poder grabar en cinta un programa ensamblado (por ejemplo la subrutina SAVE de ROM), para poder luego utilizarlo como un programa en fuente para el GENS3, y hacer las modificaciones que parezcan oportunas.

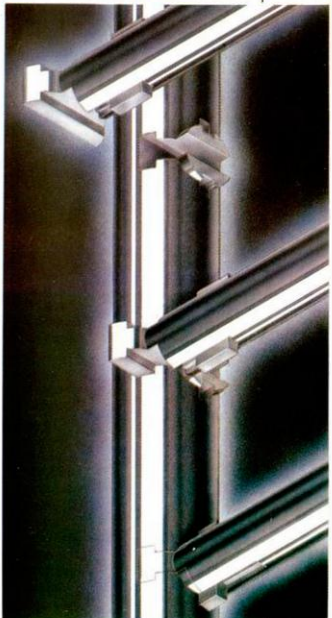
Para utilizar esta opción habrá que hacerse lo siguiente:

Primero: Hay que ver la subrutina que se va a sacar por cinta, y saber su comienzo y final.

Segundo: Después de saber los dos datos anteriores, pulsar la T. Aparecerá el mensaje «First:» al que responderemos con el inicio del bloque a copiar (4C2). Después nos preguntará «Last:» y digitaremos la dirección final (555). Después preguntará «Printer». Si respondemos Y nos dará el listado en fuente por impresora y si respondemos N continuará y nos preguntará «Text:», respondiéndole el comienzo de donde queremos que nos coloque el código fuente. Después nos preguntará «Workspace:» y le daremos una posición de memoria en donde empezará a colocar las etiquetas del código fuente. En estas dos últimas preguntas hay que tener cuidado de no introducir posiciones de memoria que estén usadas en ese momento y que destruirían los datos almacenados en ellas. Lo introducimos y nos hará otras dos preguntas más: «First:» en donde pondremos el comienzo de la zona de datos (DEFB, DEFW, etc.) en el código objeto y «Last:» en donde introduciremos el final de esta zona. Si respondemos a estas dos preguntas con ENTER, nos saldrá el listado del código fuente del programa.

Al final del listado fuente nos saldrá «Text end=» y un número que indicará el final del programa fuente.

Para grabarlo en cinta, volveremos al BASIC y teclearemos: SAVE «nombre» CODE (comienzo que hemos puesto), (final-comienzo). De esta forma, después podremos cargar este CODE en el GENS3; y cuando lo hayamos cargado, antes de empezar a usarlo, habrá que pulsar C, con lo que ya estará listo para su uso, y podremos hacer modificaciones como si el programa lo hubiéramos tecleado nosotros.



EL «C», c... erca de la máquina

David SOPUERTAS

Cuando hablamos de lenguajes de programación para los ordenadores personales la mayoría de nosotros piensa en el BASIC y alguno, los más expertos, piensan también en los lenguajes ENSAMBLADORES o en el código máquina. Pero, ante el auge que están tomando este tipo de ordenadores, las grandes empresas de SOFTWARE están desarrollando otros lenguajes de programación —PASCAL, C, LOGO, etc.— y no vemos muy lejano el día en el que los lenguajes destinados a los grandes ordenadores, como el FORTRAN e incluso el COBOL —no nos llaméis ilusos— puedan ser utilizados por nuestros ordenadores caseros.

De todos ellos, nos ha llegado con gran fuerza uno importado de los Estados Unidos. Uno del que todavía poseemos muy poca información aquí, sobre todo, en nuestro mundo de los ordenadores personales.

Nuestra intención es iniciaros en su conocimiento y contaros un poco sobre las diferencias existentes entre este lenguaje, concebido para codificar sistemas en explotación —sistemas operativos, bases de datos, etc.— y nuestro cotidiano BASIC.

La primera diferencia con la que nos encontramos es la forma en la que se genera un programa ejecutable.

El BASIC es un lenguaje de programación que ha sido diseñado para que sea fácil de aprender y de utilizar. Es un lenguaje "INTERPRETADO". ¿Qué quiere decir esto? Cuando escribimos una instrucción BASIC podemos obtener una respuesta inmediata a lo que estamos tecleando en el ordenador. ¿Por qué? Cada instrucción de nuestro programa es traducida o INTERPRETADA, por el programa encargado de traducirla que hay en el Sistema Operativo del ordenador, aisladamente del resto de las instrucciones. Cada línea es convertida a lenguaje máquina instrucción por instrucción cada vez que corre el programa.

Esto tiene una ventaja a la hora de aprender el lenguaje y depurar los programas. Cada vez que escribimos una línea de programa nos da un mensaje del estilo de:

SINTAX ERROR LINE 8010

si la línea está mal escrita sintácticamente o cualquier otro tipo de error en ejecución. Así la depuración se hace bastante sencilla.

Por otro lado, como esta INTERPRETACION ocurre cada vez que corremos nuestro programa BASIC, su ejecución es mucho más lenta que la de un programa similar escrito en otro lenguaje.

El lenguaje "C" ha sido diseñado para la codificación de sistemas así como para la programación de las utilidades más corrientes. En este tipo de aplicaciones se necesita que la ejecución de sus programas sea rápida y eficiente. Esta eficiencia se logra al ser el "C" un lenguaje PREPROCESADO y COMPILADO.

Los pasos para generar un programa ejecutable en lenguaje "C" son:

— Editar el PROGRAMA FUENTE en instrucciones "C" con el EDITOR suministrado por el fabricante para nuestro ordenador.

— Pasar este programa fuente por el PREPROCESADOR. El PREPROCESADOR "C" es una herramienta a la que llamamos sitemáticamente después de editar nuestro programa "C" y antes de llamar al COMPILADOR. Nos permite efectuar inclusiones de ficheros, elementos de la Librería Estándar del lenguaje, expandir macroinstrucciones, redefinir nombres y constantes, compilación condicional, etc. Algunas de estas facilidades no están contempladas en el "C" implementado por HISOFT, como veremos más adelante.

Las instrucciones a este PREPROCESADOR comienzan por # y están incluidas en el programa fuente.

— COMPILAR este programa preprocesado. Esto es, traducir cada una de las instrucciones en lenguaje "C" a varias instrucciones en lenguaje o código máquina específicas de nuestro ordenador. Este paso se hace a TODO EL PROGRAMA a la vez y por tanto se genera un PROGRAMA OBJETO en código máquina.

— MONTAR en el programa compilado todas las rutinas de la Librería Estándar que necesita nuestro programa y crear así ya nuestro PROGRAMA EJECUTABLE.

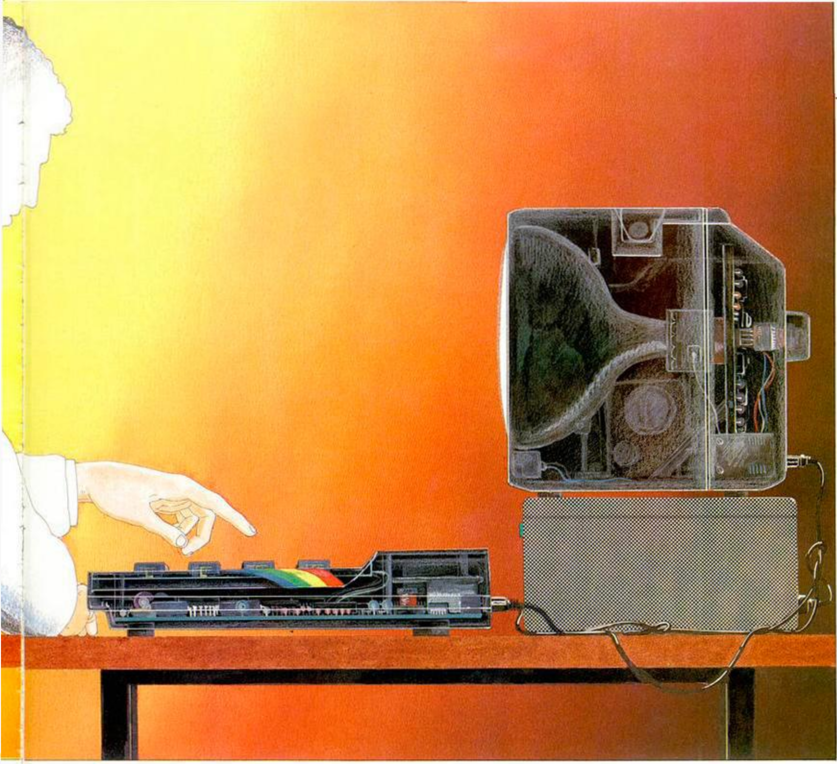
Como podemos suponer, tras todos estos pasos, es muy difícil depurar un programa ya que los cambios se han de realizar en el programa fuente y luego hay que volver a repetir todo el proceso completo. Pero tiene la ventaja que después de todo esto, el programa se ejecuta, «corre», muy rápido: es muy eficaz.



La segunda diferencia con la que nos encontramos es la estructura del programa en ambos lenguajes.

Un programa «C» consta de un conjunto de definiciones de datos y de un conjunto de procedimientos. Estos componentes no son fijos ya que un programa puede consistir solamente en una declaración de datos y un programa único, llamado «main», que indica dónde debe comenzar la ejecución del programa.

Probemos a editar el siguiente programa utilizando el EDITOR/COMPILADOR adecuado para nuestro ordenador:



```
/* PROGRAMA I */

main()
    {
     int x,y,z;
     printf("\n INTRODUCE EL PRIMER
     scanf("\n %d",&x);       NUMERO ");
     printf("\n INTRODUCE EL SEGUNDO
     scanf("\n %d",&y);       NUMERO ");
     z=mcd(x,y);
     printf("\n %s %d %s %d %s %d
    \n","EL MCD DE",
    x,"Y",y,"ES",z);
    }
/*  PROGRAMA SECUNDARIO - 2   */
/* mcd(P1,P2) */
/* FUNCION QUE CALCULA EL MCD */
/* DE DOS NUMEROS (P1 Y P2) Y */
/* DEVUELVE SU VALOR AL PRO-  */
/* GRAMA PRINCIPAL.           */
/* SI SE USA LA RECURSIVIDAD  */
mcd(a,b)
int a,b;
    {
     if (b==0)
      return a;
     else    /* RECURSIVIDAD */
      return mcd(b,a%b);
    }
#list*
```

Como vemos, el «C» es un lenguaje de formato libre en el que todas las instrucciones terminan en punto y coma. Al comienzo del programa debemos poner unos comentarios que nos definan el nombre del programa y para que sirve. Estos comentarios tendrán la forma:

/* COMENTARIOS */

A continuación va el programa principal o «main» que, como hemos dicho antes, es el punto por donde va a comenzar a ejecutarse el programa, seguido de la serie de funciones que forman parte del mismo —en este caso mcd.

Todos los datos que se utilicen han de estar declarados asociando a un identificador —nombre— su tipo correspondiente. Ocasionalmente podemos darle un valor inicial.

«TODAS LAS VARIABLES HAN DE ESTAR DEFINIDAS ANTES DE SU UTILIZACION».

Las sentencias compuestas, o grupos de sentencias están señaladas con llaves de apertura y de cierre. Un programa "C" siempre empieza con una llave abierta «{» y termina con una llave cerrada «}».

Las llamadas a funciones externas se realizan indicando el nombre de la función seguida de todos sus parámetros encerrados entre paréntesis. En este caso, las funciones «printf» y «scanf» son las funciones de entrada y salida de datos formateados que están contenidas en la Biblioteca Estándar a la que accede el compilador. El primer argumento de «scanf» es el formato de los datos que va a leer: un valor entero. El siguiente argumento es el PUNTERO a la variable sobre la que vamos a asignar el valor leído. Este nuevo concepto, PUNTERO, consiste en asignar a una variable la dirección de otra. En este caso &x y &y contendrán las direcciones de memoria donde se encuentran las variables x e y. Hay que tener claro que estas funciones no son parte del lenguaje en sí, sino que están contenidas en la Biblioteca Estándar.

Una característica de este lenguaje es que sus procedimientos pueden ser recu-

rrentes, es decir, que pueden llamarse a sí mismos. En este caso es mcd el que se llama a sí mismo.

Otra de las peculiaridades más importantes de este lenguaje es su completa adaptabilidad a la programación estructurada consiguiéndolo al cumplir las siguientes características:

– ESTRUCTURAS DE DATOS: Los datos se definen junto a su tipo correspondiente.

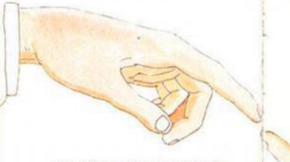
– ESTRUCTURAS DE CONTROL: Emplea solamente estructuras lógicas básicas.

– SECUENCIALIDAD: No utiliza bifurcaciones incondicionales (GOTOs).

– MODULARIDAD: Divide un módulo general en pequeños programas o procedimientos que realizan una función muy concreta y especifica y que es lo más independiente posible del resto de los programas del módulo.

A continuación proponemos que se comprueben las diferencias existentes entre este sencillo programa y su equivalente en BASIC.

```
10  REM ************************
20  REM *    PROGRAMA II       *
30  REM *  Programa Basic que  *
40 REM *  calcula el mcd de   *
50 REM *      numeros         *
60  REM ************************
70 LET mcd=180
80 INPUT "INTRODUCE EL PRIMER NUMERO ";a
90 INPUT "INTRODUCE EL SEGUNDO NUMERO ";b
100 PRINT AT 11,2;"EL MCD DE ";a;" Y ";b;" ES ";
110 GO SUB mcd
120 PRINT a
130 PAUSE 100
140 PRINT #0;"QUIERES CALCULAR OTRO MCD?"
150 IF INKEY$<>"" THEN GO TO 150
160 IF INKEY$="" THEN GO TO 160
170 LET z$=INKEY$
180 IF z$="S" OR z$="s" THEN RUN
190 STOP
200  REM ************************
210 REM *  Rutina que calcula *
220 REM * el mcd de 2 numeros *
230  REM ************************
240 IF a=0 THEN LET a=b: RETURN
250 IF a=b OR b=0 THEN RETURN
260 IF a>b THEN LET a=a-b: GO TO 250
270 LET b=b-a: GO TO 250
```



Observemos la diferencia que hay en la estructura de los programas y sobre todo el diferente tiempo de ejecución entre uno y otro. Dándoles unos valores apropiados podemos ver que el tiempo de ejecución del programa BASIC puede llegar a ¡DESESPERARNOS!

Y ahora vamos a introducirnos de lleno en las características propias del lenguaje «C».

Declaraciones de datos

El lenguaje «C» tiene unos tipos básicos que pueden definir datos. Nos definen también el rango que una variable puede tomar así como un tamaño de su representación en la memoria.

Los tipos básicos en C son:

– Caracter:

char caracter;

define la variable «caracter» de tipo caracter.

– Entero:

int x;

Este tipo puede admitir los calificadores «short» y «long» que definen enteros de distinta longitud y «unsigned» que define a los números enteros positivos.

– Reales:

float y;
double z;

nos definen dos números reales de los que «y» es de coma flotante y precisión simple y «z» es una variable





de coma flotante y doble precisión.

Lo mismo que en BASIC podemos definir una variable literal como «variables$», en «C» no existe la noción de cadena como tipo definido. Para menjar las cadenas tenemos que utilizar el concepto de PUNTERO antes visto. Se definirían:

char * cadena;

donde cadena es una variable de tipo PUNTERO que apunta al primer caracter de la cadena.

Una vez que hemos visto como se pueden definir las variables asociándolas con su tipo, hablaremos del ámbito o zona de un programa donde un nombre de una variable es conocido con un significado determinado.

En «C» hay dos tipos de ámbitos:

– Variables externas o globales. Son las variables que están declaradas fuera de funciones. Una variable externa es reconocida desde su declaración hasta el final del programa fuente. Además pueden ser accedidas desde otras fuentes dentro del sistema.

Aquí encontramos otro calificativo «extern» que indica que las variables ya han sido declaradas en otro programa.

– Variables internas. Son las variables que están declaradas en el interior de las funciones. Pueden ser:

AUTOMATICAS: Se crean cuando se llama a la función en la que están definidas. A la salida de esta función, la variable pierde su valor.

int número;

ESTATICAS: Llevan el atributo «static». Siguen siendo variables particulares de una función, pero se las asigna almacenamiento permanente y por lo tanto no pierden su valor.

static char letra;

Cuando preveemos que una variable va a tener un uso muy frecuente, el «C» nos permite hacer una de las cosas más interesantes de este lenguaje: almacenar una variable en uno de los registros de la máquina.

La declaración se hace de la siguiente forma:

register int cifra;

e indica al compilador que la variable «cifra» será frecuentemente utilizada y que la coloque en un registro de acceso rápido, lo que permite un ahorro de tiempo.

Como podemos observar, en BASIC todo esto no existe. Las variables son todas globales y por supuesto que no podemos reservar ningún registro para una que utilicemos frecuentemente.

Operadores

Además de los operadores aritméticos, relacionales y lógicos, el lenguaje «C» está dotado de una serie de operadores de bajo nivel —semejante al lenguaje «ENSAMBLADOR»— que le dan una gran potencia.

Estos operadores pueden ser:

– Operadores de incremento-decremento:

Son los equivalentes a las instrucciones DEC e INC del lenguaje «ENSAMBLADOR».

x++ ++x

y— —y

En ambos casos se incrementa la variable x o se decrementa la variable y pero según dónde se encuentren los signos la operación se

realizará antes o después de que la variable sea utilizada dentro de la expresión de la que forma parte.



```
/* PROGRAMA IV */
main()
(
 int x,y;
     x=204;
     y=32;
     printf("Hexadecimal X = 0x%x  Y = 0x%x\n",x,y);
     printf("\nX desplazado a la izquierda 3   bits = 0x%x\n", x<<3);
     printf("\nY desplazado a la derecha 3 bits = 0x%x\n",y>>3);
     printf("\nX OR Y = 0x%x\n",x|y);
     printf("\nX AND Y = 0x%x\n",x&y);
)
```

El Programa III nos intentará despejar las dudas que tengamos al respecto.

```
/* PROGRAMA III */
main        ()
(
 int x,y;
   x=y=40;
   printf("\n++X = %d  Y++ = %d",++x,y++);
   printf("\nX = %d    Y = %d\n",x,y);
   printf("\n--X = %d  Y-- = %d",--x,y--);
   printf("\nX = %d    Y = %d",x,y);
)
```

Estos operadores sólo pueden ser aplicados sobre variables, NUNCA SOBRE EXPRESIONES.

— Operadores lógicos sobre bits: son operadores que realizan sus operaciones lógicas manipulando con los BITS.

Son los siguientes:

&: operador lógico AND.

|: operdor lógico OR.

/ : operador lógico XOR.

<<: desplazamiento de bits a la izquierda. Rellena los huecos dejados con ceros.

>>) desplazamiento de bits a la derecha. Generalmente también rellena los huecos con ceros.

^: complemento a 1.

Un pequeño ejemplo sobre este tipo de ordenadores lo tenemos en el Programa IV.

Arrays, punteros y estructuras

A los tipos de datos vistos anteriormente hay que añadirles otros tipos derivados que podemos definirlos como matrices (arrays), punteros y estructuras.

Un ARRAY es una colección de datos del mismo tipo, que están referidos por el mismo nombre.

Se definen de la siguiente manera:

— Arrays unidimensionales:

char nomber[5O];
int tabla[12O];

— Arrays multidimensionales:

int matriz [3] [7] [1O];
static int matriz [4] [7] [2O];

Los elementos individuales dentro de un array pueden referenciarse con un indice, que debe ser un dato entero o una expresión compleja pero cuya evaluación ha de darnos un dato entero, y determinar el número de datos del tipo básico que forman parte del array.

PUNTERO es una variable que contiene la dirección de otra variable.

punterov = &vector;

asigna la dirección de la variable «vector» a la variable «punterov».

y = •punterov;

da a la variable «y» el valor del contenido de la variable a la que apunta la variable «punterov».

El resultado de las dos instrucciones anteriores sería equivalente a:

y = vector;

Intentemos analizar el siguiente Programa:

```
/* PROGRAMA V */
main()
(
 int vector[10];
 int *pv;
 int n;
/* Dar valores a los
elementos de la matriz*/
     for (n=0;n<10;n++)
      vector[n]=n;
/*Indicacion de fin
del array */
     vector[n]=-1;
/*Dar la direccion de
vector al puntero pv*/
     pv=vector;
/*Escribir cada uno de
los elementos de la
matriz a la que
apunta el puntero pv*/
     while (*pv0)
      (
       printf(" %d",*pv);
       pv++;
      )
     putchar('\n');
)
```

En este pequeño programa vemos que •pv apunta al primer elemento de la matriz vector, •(pv+1) apunta al segundo elemento y así sucesivamente. Su equivalente en BASIC sería recorrer con un indice toda la tabla escribiendo cada elemento cada vez que accedemos a él.

```
100 DIM vector(10)
110 FOR i = 0 TO 9
120 PRINT vector(i)
130 NEXT I
```

pero este proceso es generalmente más lento que la aplicación de los punteros.

Otro tipo derivado utilizado en el lenguaje «C» son las ESTRUCTURAS. Son agrupaciones de variables, que pueden ser de distinto tipo básico, agrupadas bajo un nombre único para facilitar su manejo.

Un ejemplo de definición de estructuras es:

```
struct fecha {
        int dia;
        char mes[3];
        int año;
        }
```

Las variables de este tipo se definirán:

struct fecha x;

La forma de referenciar a esta estructura es haciéndolo por su nombre x o dirigiéndose a uno de sus campos en concreto:

x.año = 1985;

Podemos utilizar también los PUNTEROS para trabajar con miembros de ESTRUCTURAS.

El modo de referenciar uno de los elementos de la estructura «fecha» mediante punteros sería:

a) Definir en el procedimiento el puntero

struct fecha •puntero;

b) Referenciar uno de los miembros

puntero —> dia;
puntero —>[1];

Con los comentarios del Programa VI intentemos entender todo lo anteriormente dicho sobre ESTRUCTURAS.

La función «strcpy», utilizada en él, copia una cadena de caracteres en otra y está incorporada a la Biblioteca Estándar del sistema. Por esto empleamos los comandos #include «stdio.h» y # include ?stdio.lib? para que el PREPROCESADOR incluya las declaraciones y las funciones de la Librería Estándar que este programa necesita.



```
#include "stdio.h"
/* PROGRAMA VI */
struct empleado(
          int clave;
          char nombre[50];
          );
main()
(
 struct empleado em;
/* Ponemos en clave 100 */
     em.clave=100;
/*Escribimos en nombre la
cadena "nombre-empleado"*/
    strcpy(em.nombre,"nombre de empleado");
/*Sacamos el primer
mensaje con los datos
que tenemos*/
     printf("PROGRAMA: \nCLAVE %d
\nNOMBRE %s",
em.clave,em.nombre);
/*Llamada al programa
que cambia los datos
*/
     copiar(&em);
/*Mensaje con los
nuevos datos */
     printf("\nPROGRAMA: \nCLAVE %d
\nNOMBRE %s", em.clave,
em.nombre);
)
/* Procedimiento copiar*/
copiar(registro)
 struct empleado *registro;
(
     printf("\n FUNCION: \nCLAVE %d
\nNOMBRE %s",
registro->clave,registro->nombre);
/* Da nuevos valores
a los campos de
la estructura sobre
la que apunta la
variable registro*/
     registro->clave=300;
     strcpy(registro->nombre,
"Otro nombre de
empleado");
)
#include ?stdio.lib?
```

Estructuras de control

El lenguaje «C» nos ofrece estructuras de control que nos permiten codificar empleando estructuras básicas que responden a las características requeridas por la programación estructurada, con las que se mejora la claridad de los programas. Son bastante parecidas a las de lenguajes de alto nivel, como el «PASCAL».

If-Else

Esta estructura es muy clásica en casi todos los lenguajes, e incluso aparece en algunos «BASICs» evolucionados.

Su formato es:

```
if (< expresión >)
  |
     acción–1;
  |
else
  |
     acción–2;
  |
```

Si se cumple la condición evaluada en la expresión, el programa realiza la acción–1, sino, realiza la acción–2.

El ELSE es opcional y siempre esta asociada al IF inmediatamente anterior.

While

Su estructura general es:

```
while (< expresión >)
     |
       acción;
     |
```

Mientras se cumpla la expresión, se ejecutan las instrucciones que forman parte de la acción.

Si lo que queremos es que una serie de instrucciones se ejecuten, basta que se cumpla una condición. Utilizaremos la estructura DO WHILE de la siguiente forma:

```
do
 |
   acción;
 |
   while (< condición >);
```

```
/* PROGRAMA VII */
main()
(
 int x,y;
    x=10;
    do
/* Este bucle se repite
   hasta que x sea cero */
     (
      y=x;
      do
/* Este bucle interno se
   repite hasta que y sea
   menor que cero */
       (
        printf(" %d",y);
        y--;
       )
       while (y0);
      putchar('\n');
      x--;
     )
    while (x!=0);
)
```

El cuerpo del bucle DO WHILE se ejecuta al menos una vez, ya que la evaluación de la condición se efectúa al final del bucle.

Un ejemplo de esto sería el Programa VII.

Bucle for

El formato más general de esta sentencia es:

```
for ( condición–1; condición–2; expresión)
       |
         acción;
       |
```

en el que condición–1 es la condición inicial; condición–2 es la de salida del bucle y expresión suele ser la actualización de la variable de control del bucle.

Es una de las instrucciones más potentes del lenguaje y por ello nos permitimos sugerir que se trabaje intentando hacer variaciones sobre este pequeño ejemplo.

```
/* PROGRAMA VIII */
main()
(
 int i, j;
     for(i=0;i(5;i++)
      (
       for(j=0;j(5;j++)
        printf("   %d,%d",
                i,j);
       putchar('\n');
      )
)
```

Switch

Permite realizar de una manera más clara una cadena de sentencias IF... ELSE IF... ELSE.

El formato general de esta sentencia es:

```
switch ( expresión )
 |
    case <constante–1>:
    acción–1;
    case <constante–2>:
    acción–2;
    .
    case <constante–n>:
    acción–n;
    default: acción–n+1;
```



En su realización, la expresión evaluada se compara con cada una de las constantes.

Si no colocamos una instrucción BREAK al final de cada caso, se ejecuta el resto del SWITCH. Esta instrucción BREAK hace que se abandone el SWITCH inmediatamente.

Para reafirmar esto usaremos el siguiente Programa:

(Ver programa IX)

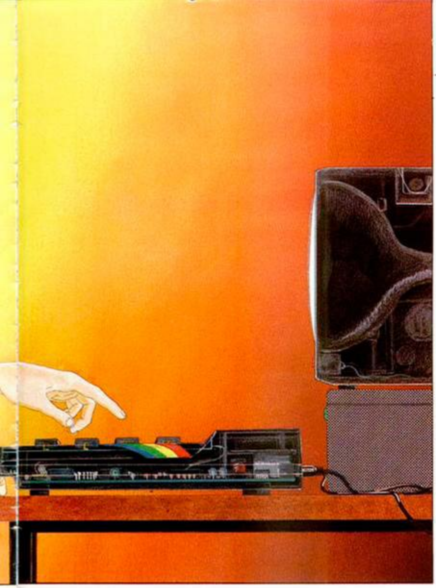
Break y Continue

BREAK, como hemos dicho antes, es una instrucción que hace que se abandone el bucle más interno o el SWITCH.

Da a los bucles un punto más de salida y para a la siguiente sentencia del programa, aunque se siga cumpliendo la condición del bucle.

CONTINUE hace que rompamos la secuencia de ejecución de sentencias dentro de un bucle y pasemos a la siguiente iteración del bucle.

(Ver programa X)



El preprocesador C

Ya hemos tenido anteriormente una pequeña noción de lo que es el PREPROCESADOR C. Es un paso necesario a la llamada al complador. Todas sus instrucciones van precedidas por # al principio de la línea.

Principalmente permite efectuar:

— Macro-sustituciones

```
# define superior 50
```

realiza una sustitución en todos los lugares del programa donde se encuentre el nombre «superior» por una cadena de caracteres — en este caso 50.

La definición de un nombre es válida desde el momento en que se hace hasta el final del programa fuente.

Estos nombres pueden ser redefinidos en un momento determinado del programa.

```
# define min (A,B)
((a)>(B)?(B):(A))
```

llega más lejos. En este caso hemos definido una macro con argumentos.

Esta instrucción hará que la línea

```
x = min (p,q);
```

sea expandida en:

```
x = ((p)>(q)?(q)(p))
```

Es una pena que esta facilidad no la podamos utilizar con la implementación de C de HISOFT.

— Inclusión en ficheros:

```
# include fichero
```

produce una sustitución de esta línea por el contenido del fichero especificado.

— Compilación condicional (Tampoco está contemplada en HISOFT C).

```
/* PROGRAMA IX */
main()
{
 int cont_1,
     cont_2,
     cont_3,
     cont_4;
 char caracter;
     cont_1=
     cont_2=
     cont_3=
     cont_4=0;
     printf("\nESCRIBE UN NUMERO DE 1 A 3\n(ENTER PARA TERMINAR)\n");
     caracter=getchar();
     while (caracter!='\n')
       {
       switch(caracter)
       {
       case'1':cont_1++;
               printf("\n  ES UN 1\n");
               break;
       case'2':cont_2++;
               printf("\n  ES UN 2\n");
               break;
       case'3':cont_3++;
               printf("\n  ES UN 3\n");
               break;
       default:cont_4++;
                          printf("\nCHUNGO, NO ES NINGUNO DE LOS 3\n");
               break;
       }
       printf("\nESCRIBE UN NUMERO DE 1 A 3\n(ENTER PARA TERMINAR)\n");
       getchar('\n');
       caracter=getchar();
      }
     printf("\n\nRESUMEN TOTAL\n");
     printf("\nCONTADOR DE 1 = %d",cont_1);
     printf("\nCONTADOR DE 2 = %d",cont_2);
     printf("\nCONTADOR DE 3
=%d",cont_3);
     printf("\nCONTADOR DE NULOS
=%d",cont_4);
```

```
/* PROGRAMA X */
main()
{
 int x,y;
     x=10;
     while (x!=0)
      {
       if (x==8)
        {
         printf("\n\nCORTAMOS EL BUCLE EXTERNO %d",x);
         break;
        }
       else
        {
         y=x+1;
         printf("\n\nBUCLE EXTERNO %
%d\n",x);
        }
       while (y>0)
        {
         y--;
         if (y==5)
          {
           printf("\n\n  SE SALTA EL BUCLE INTERNO %d\n",y);
           continue;
          }
         else
          printf("\n  BUCLE INTERNO %d",y);
       }
       x--;
      }
}
```

```
10  REM ***********************
20  REM *     PROGRAMA II     *
30  REM *  Programa Basic que *
40 REM *  calcula el mcd de  *
50 REM *       numeros       *
60  REM ***********************

70 LET mcd=160
80 INPUT "INTRODUCE EL PRIMER NUMERO ";a
90 INPUT "INTRODUCE EL SEGUNDO NUMERO ";b
100 PRINT AT 11,2;"EL MCD DE ";a;" Y ";b;" ES ";
110 GO SUB mcd
120 PRINT a
130 PAUSE 100
140 PRINT #0;"QUIERES CALCULAR OTRO MCD?"
150 IF INKEY$<>"" THEN  GO TO 150
160 IF INKEY$="" THEN  GO TO 160
170 LET z$=INKEY$
180 IF z$="S" OR z$="s" THEN  RUN
190 STOP

200  REM ***********************
210 REM *  Rutina que calcula *
220 REM * el mcd de 2 numeros *
230  REM ***********************

240 IF a=0 THEN  LET a=b: RETURN
250 IF a=b OR b=0 THEN  RETURN
260 IF a>b THEN  LET a=a-b: GO TO 250
270 LET b=b-a: GO TO 250
```

if < expresión >

.

.

.

endif

Esto nos permite que, si la evaluación de la expresión no es cierta, el copilador ignore todas las líneas entre estas dos sentencias.

Puede aparecer también la expresión # else. En este caso el compilador ignora las líneas comprendidas entre else y endif si la evaluación de la expresión es cierta.

— Acceder a la Librería Estándar

include studio. h
include studio. lib

nos permite acceder a todas las funciones y definiciones incluidas en la Librería Estándar.

Como habéis podido observar en esta ojeada, el lenguaje «C» es un tema muy amplio e interesante. Se nos han quedado en el tintero temas como las funciones, el manejo de ficheros y el conocimiento de la Librería Estándar del lenguaje.

El «C» es ágil y moderno y tiene todas las características de un lenguaje de alto nivel, acompañado además de la eficiencia y velocidad de los de bajo nivel. Al mismo tiempo, es un lenguaje muy apropiado para la programación estructurada, dándonos unos programas muy claros y fáciles de leer y entender.

Esperamos que estas nociones que os hemos dado sirvan para despertar vuestra curiosidad sobre el lenguaje y os lanzéis a demostrar que vuestro ordenador casero es capaz de trabajar en buena armonía con el «C» sin sentir envidia de los grandes ordenadores.

MAS ALLA DEL TIEMPO, MAS ALLA DEL ESPACIO...



Dinamic Software presenta una nueva realización para Spectrum 48K · Plus · 128K. P.V.P.: 1.950 Ptas.

SGRIZAM. LA ESPADA DEL PODER

Tiendas y distribuidores Tel.: (91) 447 34 10. Pedidos contrareembolso. Tel.: (91) 715 00 67

¡¡¡Incluye operación puzzle: 6 millones de pesetas de regalo en programas. No te lo pierdas!!!